CAMAMED manual

CAMAMED: a pipeline for composition-aware mapping-based analysis of metagenomic data

Software Requirements

- Linux operating system (Preferably Ubuntu)
- Python 2.7
- MetaPhlAn2
  - Installation command
    - sudo apt install metaphlan2
  - After the first run, MetaPhlAn database files are downloaded automatically. Otherwise, download the files from the following links and copy them to the installation path in folder /usr/share/metaphlan2/databases.
    - https://bitbucket.org/biobakery/metaphlan2/downloads/mpa_v20_m200.tar
    - https://bitbucket.org/biobakery/metaphlan2/downloads/mpa_v20_m200.md5
- CD-HIT
  - Installation command
    - sudo apt-get install cd-hit
- SRA-Toolkit
  - Installation command
    - sudo apt install sra-toolkit
- Samtools
  - Installation command
    - sudo apt-get install samtools
- Also install the necessary packages for Python if requested. For example:
  - sudo apt-get install python-numpy
  - sudo apt-get install python-pandas
  - pip install biopython
- Install R package for python 2.7
  - pip install rpy2==2.8.6
  - conda install -c r r-glmnet
  - conda install -c r r-kernsmooth

- python2
- from rpy2.robjects.packages import importr
- base = importr('base')
- base.source("http://bioconductor.org/biocLite.R")
- biocinstaller = importr("BiocInstaller")
- biocinstaller.biocLite("metagenomeSeq")
- biocinstaller.biocLite("edgeR")

Hardware Requirements

This software can run on a regular PC with 8GB RAM and a single processor. Of course, it depends on the size of the samples and the gene catalog, but for a run with the higher performance, it is better to run on a computer with at least 32GB of RAM and ten cores of processor.

Before starting

- Copy the FASTA format gene catalog files in the /metagenomic_data_analysis/ folder
- If the sequence files are SRA format
  - Copy SRA samples in the /metagenomic_data_analysis/sra_files folder.
  - Copy the sample names in the /metagenomic_data_analysis/ sra_file_names.txt file.
  - for example:
    - sra_file_names:
      file1.sra
      file2.sra
      file3.sra
  - After executing SRA-Toolkit, Fastq or Fasta files are automatically copied to folder /Read_files.
- Copy Fastq or Fasta samples in the /metagenomic_data_analysis/Read_files folder (If SRA-Toolkit is not executed).
- Copy the sample names in the /metagenomic_data_analysis/sample_file_names.txt file and label of them in the /metagenomic_data_analysis/Read_files/class_label.txt file. for example:
  - sample_file_names:

p_file1_1.fq
p_file1_2.fq
p_file2_1.fq
p_file2_2.fq
p_file3_1.fq
p_file3_2.fq

- class_label:

class1
class2
class3

If the samples are paired-end, enter a label for two files that is typed in sequence.

- A sample of the gene catalog is located in the folder below:
  - /metagenomic_data_analysis/sample_files/gene_catalog
- Some samples, with their names and labels, could be found in the following folder:
  - /metagenomic_data_analysis/sample_files/Read_files
- For example, KAAS and GhostKOALA outputs are in the following folder:
  - /metagenomic_data_analysis/sample_files/gene_ko_outputs

Pipeline menu

To run this pipeline in the Linux terminal, run the following command.

- python2 mamed.py

1. Enter 'a' for Preprocessing.

1.1. Enter 'a1' for running SAR toolkit to convert SRA files to Fastq or Fasta files. (**optional**)

1.2. Enter 'a2' to receive information on sequences and gene catalogs.

1.3. Enter 'a3' for running CD-HIT on the gene catalog to remove redundant genes (**optional**).

1.4. Enter 'a4' for preprocessing gene catalog.

2. Enter 'b' for FastQC quality control.

2.1. Enter 'b1' to execute FastQC quality control only for Fastq files.

2.2. Enter 'b2' to execute SeqKit to extract information from sample files.

2.3. Enter 'b3' to extract total information from SeqKit outputs.

3. Enter 'c' for MetaPhlan2.

3.1. Enter 'c1' to execute metaphlan2.

3.2. Enter 'c2' to extract information from metaphlan2 output files.

4. Enter 'd' for Mapping Reads to reference gene catalog using MOSAIK.

4.1. Enter 'd1' for creating an acceptable form of gene catalog.

4.2. Enter 'd2' for mapping reads to gene catalog.

5. Enter 'e' for extracting samples information using the user's gene catalog and KEGG database.

5.1. Enter 'e1' for preparing the Gene Catalog for extracting annotated information from the KEGG databases.

5.2. Enter 'e2' to extract information from GhostKOALA or KAAS output files.

5.3. Enter 'e3' to extract annotated information that has already been extracted from the KEGG database.

5.4. Enter 'e4' to extract the annotated information from the KEGG database (**optional**).

5.4.1. Enter 'e41' to extract KO-related EC numbers from the KEGG database.

5.4.2. Enter 'e42' to extract EC-related reactions from the KEGG database.

5.4.3. Enter 'e43' to extract reaction-related equations from the KEGG database.

6. Enter 'f' for Extracting samples information from normalized bacteria, gene, KO, EC number and reaction matrix.

6.1. Enter 'f1' for normalizing data using local information previously extracted from the KEGG database.

6.2. Enter 'f2' for normalizing data using information extracted by the user from the KEGG database.

7. Enter 'g' for performing statistical KRUSKAL-WALLIS test on normalized data.

7.1. Enter 'g1' for running the KRUSKAL-WALLIS test on the default annotated data.

7.2. Enter 'g2' for running the KRUSKAL-WALLIS test on the user extracted data.

Important points

Point1: Note that if you have a storage limitation to store the files related to entire sequences, you can copy permitted number of files in the /sra_files/ or /Read_files/ folder and save the names of the copied files in 'sra_file_names.txt' or 'sample_file_names.txt' files and processes a1, b1, b2, c1 and d2 on the copied sequences.

Point2: Steps a2, a3, a4, and d1 are performed on the gene catalog, and in any case, should be done at the beginning of the work.

Point3: After running Point1 on all sequences and Point2, enter the entire sample's name in the 'sample_file_names.txt' file as described in the 'Before starting' section and run the remaining steps that are related to the whole sequences.

But if you do not have the storage limitation, copy all sequence files in the order and execute the steps in sequence

Commands description

1.1. 'a1'

If the sequence format is SRA, then enter the files in the /sra_files/ folder first and enter them as described in the 'Before starting' section and select this option. After executing this section, the SRA files are converted to Fastq or Fasta files and copied to the /Read_files/ folder. After running this step, enter the file names in the text file 'sample_file_names.txt' as described in the 'Before starting' section.

1.2. 'a2'

At this point, information about the gene catalog, sequence samples, and processing information must be entered. The following questions should be answered in sequence

Gene catalog name:

  - for example: gene_catalog.fa or gene_catalog.fasta
- Type of sequences:
  - fastq or fasta
- Type of read files
  - p or s for paired-end or single-end, respectivelly
- Number of cores:
  - 1 to n
- Insert size for paired-end sequencing (to ignore insert size, enter -1)
  - For example : 300

1.3. 'a3'

Using of this option is optional. You can use this option if the gene catalog is constructed by the user or if you want to delete the redundant sequences. To continue, you must select the sequence identity threshold for redundant sequences in the permitted interval [0.8: 1], with a default value of 0.9.

After deletion of redundant genes, the new gene catalog is saved with the name cd_hit_gene_catalog. Also, clustered genes are saved in a file named cd_hit_gene_catalog.clstr, and the genes of the head cluster are marked with asterisk (*).

1.4. 'a4'

At this step, the names of the genes are deleted from the gene catalog and stored in the /metagenomic_data_analysis/gene_name.txt file, and for the genes, the gene1, gene2, etc. are respectively selected. Finally, the gene catalog saved with the name /metagenomic_data_analysis/main_gene_catalog.fa

2.1. 'b1'

This option is used to control the quality of Fastq sequences, and if the sequences are Fasta, this option is unavailable. In this step, FastQC software executes on sequences and stores the output as HTML files in the /metagenomic_data_analysis/fastqc_output/ folder.

2.2. 'b2'

At this point, SeqKit software is run to extract the statistical information of the samples, and saved in the /metagenomic_data_analysis/seqkit_output folder.

2.3. 'b3'

In this step, all the statistical information related to the Seqkit outputs is extracted and saved in the /metagenomic_data_analysis/all_results/total_sample_info.txt file.

3.1. 'c1'

By choosing this option, the MetaPhlan2 software runs on samples, and the results are stored in the /metagenomic_data_analysis/metaphlan_output folder. MetaPhlan2 can produce taxonomic profiling at different levels Such as Kingdom, Phylum, Class, Order, Family, Genus, or Species. To do this, you must select 'k', 'p', 'c', 'o', 'f', 'g' or 's' respectively. But it is recommended to use family, genus, or species level.

You can use the metaphlan2.py command to access MetaPhlan2 help. Meanwhile, MetaPhlan2 only returns the results of prokaryotic genomes and ignores the genomic information of eukaryotes, viruses, and archaea. For more configurations, refer to the /metagenomic_data_analysis/metaphlan_samlpe.sh file.

3.2. 'c2'

At this step, information about the bacteria is selected at the taxonomic level, and their frequency is calculated (Frequency is reported as percentages) and stored in the /metagenomic_data_analysis/all_results/total_metaphlan_results.txt file. If the sequences are paired-end, instead of the two files per one sample, only one output is reported as an average.

4.1. 'd1'

At this stage, two MosaikBuild and MosaikJump tools are run on the gene catalog to prepare it for sequence mapping. To access the help of these tools, you can run './MosaikBuild -h', and './MosaikJump -h' commands in the Linux terminal at /metagenomic_data_analysis/ path, and you can refer to the /metagenomic_data_analysis/mosaik_build_ref.sh file for further configuration. Also, refer to the https://github.com/wanpinglee/MOSAIK/wiki/QuickStart link for further details. The most important parameter at this state is the length of the hash word, which can be selected in the interval of [4:32], and its default value is 15. For more configurations, refer to the /metagenomic_data_analysis/mosaik_build_ref.sh file.

4.2. 'd2'

At this state, the MosaikBuild tool is used for preparing the sequences and the MosaikAligner tool is used to map the sequences to the gene catalog prepared in step 4.1. To access the help of these tools, the './MosaikBuild -h' and './MosaikAligner -h' commands are executed at /metagenomic_data_analysis/ path. Refer to the /metagenomic_data_analysis/mosaik_read_aligner.sh file for more configurations. The mapping results are stored in a SAM format file in the /metagenomic_data_analysis/mosaik_outputs folder.

5.1. 'e1'

To extract the annotation of the gene sequences from the KEGG database, we will use both of the KAAS and the GhostKOALA web services. But the size of the gene catalog should be less than 300MB. Otherwise, you can use the option of converting the gene catalog to several smaller files. For instance, if a gene catalog possesses 300 genes, enter the parameters for this splitting option as '100 200'. This will split the gene catalog into three files, each with 100 genes. If the file size does not get smaller than 300MB, this step should be re-run. Finally, the smaller files are located in the /metagenomic_data_analysis/sub_catalog_files folder and are ready to be uploaded to the web service.

5.2. 'e2'

After the conversion of the gene catalog into the files smaller than 300MB, for nucleotide sequences, both the KAAS and the GhostKOALA web services can be used to obtain KOs associated with each gene sequences. However, for amino acids, only GhostKOALA can be used to get KOs.

- Link to the KAAS web service for uploading sequences
  - https://www.genome.jp/kaas-bin/kaas_main
- Link to the GhostKOALA web service for uploading sequences
  - https://www.kegg.jp/ghostkoala/

Two examples of the KASS and the GhostKOALA web services output are in the /metagenomic_data_analysis/sample_files/gene_ko_outputs folder. If the gene catalog is more than one file, the results should also be saved in a few text files, and the order in which they will be uploaded to the software will be the same as the order of gene number. To continue running, Web services output files must be copied to the /metagenomic_data_analysis/kegg_annotation folder. At this point, the names of the files should be entered, for example, 'file1.txt file2.txt'. The ordering of the files is based on the order of gene number.

After this step, two ko.txt and gene_ko.txt files are created in the /metagenomic_data_analysis/kegg_annotation folder, in which the entire KOs in the samples and the relationship between the genes and the KOs are determined, respectively.

5.3. 'e3'

The following information is extracted from the KEGG database and stored in various files in the /metagenomic_data_analysis/kegg_annotation folder. This information is extracted from the Automatic Annotation Server Ver. 2.1 on 2019/6/30.

EC numbers related to KOs and all EC numbers in the KEGG database are stored in def_ko_ec.txt and def_ec.txt files, respectively.

- Reactions related to EC numbers and all reaction numbers in the KEGG database are stored in def_ec_re.txt and def_re.txt files, respectively.
- Finally, the reaction definitions and the equation for each reaction are stored in the def_re_eq.txt file

After this step, two other files are created, and the EC numbers associated with each gene and the reactions associated with each gene are stored in gene_ec.txt and gene_re.txt files, respectively.

5.4. 'e4'

If you do not want to use the previously extracted data from the KEGG database, you can use this step instead of step 5.3. There are three options at this stage to be executed in sequence.

At this state, annotated information will be extracted online from the KEGG database, but the implementation of this step will be time-consuming (Running of this step is optional).

5.4.1. 'e41'

At this step, all EC numbers associated with the ko.txt file are extracted online from the KEGG database and stored in separate files called ec.txt and ko_ec.txt. Also, the relationship between genes and EC numbers is stored in the gene_ec.txt file. Before the execution of this step, step 5.2 should be implemented.

5.4.2. 'e42'

At this step, all reactions associated with the ec.txt file are extracted online from the KEGG database and stored in separate files called re.txt and ec_re.txt. Also, the relationship between genes and reactions is stored in the gene_re.txt file.

5.4.3. 'e43'

At this state, the definitions and equations for the reaction of the re.txt file are extracted online from the KEGG database and stored in the re_eq.txt file.

6.1. 'f1'

At this state, the abundance of bacteria and genes, as well as KOs, EC numbers and reactions that are related to the identified genes in the previous sections, are normalized based on the metagenomseq1 algorithm presented in the main paper and stored in the /metagenomic_data_analysis/all_results folder. The names of these files are normal_matrix_metaphlan.txt, normal_matrix_gene.txt, normal_matrix_ko.txt, normal_matrix_ec.txt and normal_matrix_re.txt respectively. In this section, all normalizations are based on the default data extracted in section 5.3.

6.2. 'f2'

This step is exactly like Section 6.1, except that the normalized data is extracted based on the extracted online information in Section 5.4, and the same files are generated in the /metagenomic_data_analysis/all_results folder.

7.1. 'g1'

At this state, the Kruskal-Wallis test is performed on the normalized information in section 6.1. To execute this state, the label of the samples should be specified according to the description given in the section 'Before starting'. For example, to run an KRUSKAL-WALLIS test on a gene, the inputs are the normalized frequency of that gene in different samples and label of samples.

The only parameter that is specified in this step is the *p*-value to filter the output, which can be in the range [0: 1]. Finally, the KRUSKAL-WALLIS test results are stored based on the selected p-value in the /metagenomic_data_analysis/all_results folder. The names of these files are Kruskal-Wallis _test_metaphlan.txt, Kruskal-Wallis _test_gene.txt, Kruskal-Wallis _test_ko.txt, Kruskal-Wallis _test_ec.txt, and Kruskal-Wallis _test_re.txt, which have been generated for bacteria, gene, KO, EN number, and reactions, respectively.

7.2. 'g2'

This step is exactly like section 7.1, with the difference that the KRUSKAL-WALLIS test is performed on the normalized data in section 6.2, and ultimately, the same outputs are generated with the same name.