MeTarEnz Documentation

MeTarEnz (metagenomic targeted enzyme miner) is a multi-service software that enables the targeted screening of high-throughput metagenomic data with user-defined databases and bit-score cut-offs. Moreover, MeTarEnz contains an assembly module making it capable of providing its screening services while accepting various types of input data such as raw reads, assembled contigs, and translated protein sequences. As a complementary section of this software, specially designed for a lipase mining project, trained regression models for prediction of lipases' temperature and pH optima are included as well. This tool is freely accessible in forms of the standalone toolkit, docker image, and python package. Figure 1 is a pictorial workflow of MeTarEnz's services.

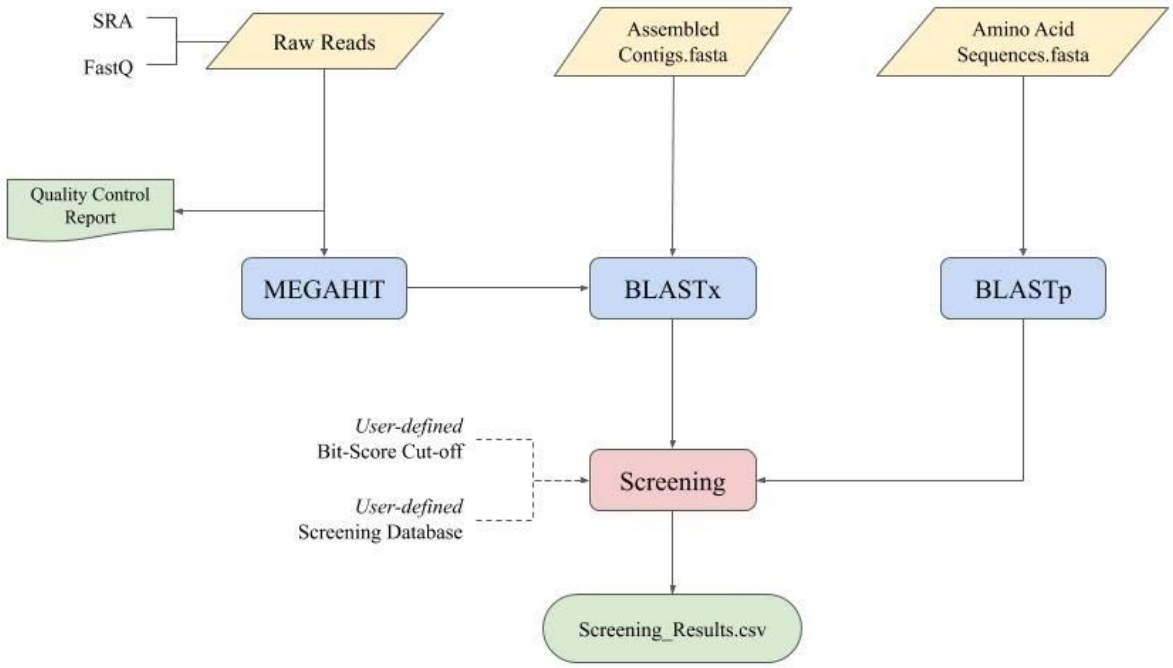

Figure 1: Workflow of Services. This figure illustrates the general workflow of the MeTarEnz including its acceptable inputs, utilized software, user-defined parameters, etc.

Distributions

- Pre-built Docker Image

Firstly, to use MeTarEnz's pre-built Docker image, you must have docker software installed (https://docs.docker.com/engine/install/).

To download the MeTarEnz image, run the following command:

docker pull mforooz/metarenz

After pulling the image, you can run MeTarEnz through the command below:

docker run -ti --entrypoint "" mforooz/metarenz python metarenz.py

Or you can run a general test to check if it works fine by the following command:

docker run -ti --entrypoint "" mforooz/metarenz python test.py

For more detailed instructions on how to use docker images, containers, etc. please read the Docker documentation (https://docs.docker.com/).

- Standalone Toolkit

MeTarEnz is also available in the form of pre-built executables for linux64. The Standalone distribution of this tool can be downloaded from cbb.ut.ac.ir/metarenz or https://github.com/mehdiforoozandeh/metarenz.

- Python Package

All MeTarEnz's codes, dependencies, models, etc. are available as a python package. This version enables users to modify and use this tool according to their needs by tweaking the source code, updating dependencies, and models. The python package of this tool can be downloaded from both cbb.ut.ac.ir/metarenz and https://github.com/mehdiforoozandeh/metarenz.

General Usage Guide

Throughout this section, it is assumed that the MeTarEnz is executed from its main directory. If otherwise, it is recommended to use files' full addresses.

- MeTarEnz can also be executed interactively with -int or --interactive.

./metarenz --interactive (from the same directory of the standalone version)

python metarenz.py --interactive (from the same directory of the python package version)

docker run -ti --entrypoint "" mforooz/metarenz python metarenz.py --interactive(docker image)

In the usage guide below, for the non-interactive execution of MeTarEnz, arguments are separated with space. Figure 2 summarizes MeTarEnz's different functions, inputs, and user-defined parameters.

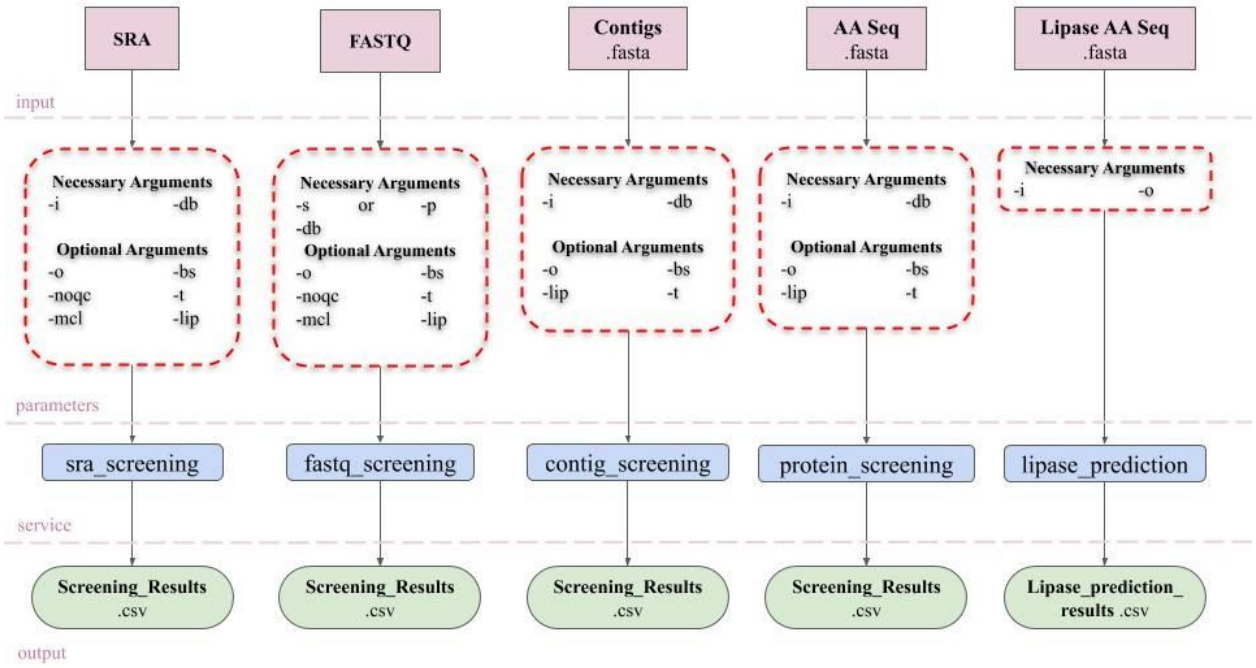

Figure 2: Summary of Functions. This figure illustrates MeTarEnz's different services, inputs, parameters, etc.

-firts argument : *./metarenz* (from the same directory of the standalone version)

python metarenz.py (from the same directory of the python package version)

docker run -ti --entrypoint "" mforooz/metarenz python metarenz.py

-second argument options:

[-h] [--help]

[-int] [--interactive]

[sras] [sra_screening]

[fqs] [fastq_screening]

[cs] [contig_screening]

[ps] [protein_screening]

[lp] [lipase_prediction]

FUNCTIONS:

[sras] or [sra_screening]: Accepts SRA file as input => assembly + quality control + screening

[fqs] or [fastq_screening]: Accepts single-ended or pair-ended fastq files as input => assembly + quality control + screening

[cs] or [contig_screening]: Accepts fasta files contining assembled contigs as input => screening

[ps] or [protein_screening]: Accepts fasta files contining protein sequences as input => screening

[lp] or [lipase_prediction]: Accepts fasta files contining lipase protein sequences as input => prediction of pH and temperature optima

HOW TO USE EACH FUNCTION:

>> [sras] or [sra_screening]:

necessary args:

[-i]: Input SRA File

[-db]: Screening Database File (*.fasta)

optional args:

[-bs]: Bit-Score Filter (Default: 50)

[-o]: Output File Name ==> Default: MeTarEnz_Results/"input_file_name"

[-noqc]: if chosen, Quality Control Results will not be reported

[-t]: Number of Threads (Default: 1)

[-mcl] or [--min-contig-len]: Minimum Contig Length (Default: 300)

[-lip]: Predict Lipase temperature and pH optima, choose only if your target enzymes are lipases

usage examples:

./metarenz sras -i srafile -db x.fasta

./metarenz sra_screening -i srafile -db x.fasta -bs 100 -o John -t 4

./metarenz sras -i srafile -db lipase.fasta -lip -bs 80 -o targeted_lipase_search -mcl 500

./metarenz sra_screening -i srafile -db x.fasta -noqc -t 5

./metarenz sras -i srafile -db x.fasta -o output -t 2 --min-contig-len 700

>> [fqs] or [fastq_screening]:

necessary args:

[-s] or [-p]: Input Fastq File => single-ended input ==>> -s file1.fq

=> paired-ended input ==>> -p file1.fq file2.fq

[-db]: Screening Database File (*.fasta)

optional args:

[-bs]: Bit-Score Filter (Default: 50)

[-o]: Output File Name ==> Default: MeTarEnz_Results/"input_file_name"

[-noqc]: if chosen, Quality Control Results will not be reported

[-t]: Number of Threads (Default: 1)

[-mcl] or [--min-contig-len]: Minimum Contig Length (Default: 300)

[-lip]: Predict Lipase temperature and pH optima, choose only if your target enzymes are lipases

usage examples:

./metarenz fqs -s fastq_file.fq -db x.fasta

./metarenz fastq_screening -db x.fasta -p fastq_file_1.fq fastq_file_2.fq -t 2 -bs 200

./metarenz fqs -p fastq_file_1.fq fastq_file_2.fq -db x.fasta -noqc

./metarenz fastq_screening -db lipase.fasta -o targeted_lipase_search -lip

./metarenz fqs -db x.fasta -s fastq_file.fq -mcl 400 -bs 200

>> [cs] or [contig_screening]:

necessary args:

[-i]: Input Contig File (*.fasta)

[-db]: Screening Database File (*.fasta)

optional args:

[-bs]: Bit-Score Filter (Default: 50)

[-o]: Output File Name ==> Default: MeTarEnz_Results/"input_file_name"

[-t]: Number of Threads (Default: 1)

[-lip]: Predict Lipase temperature and pH optima, choose only if your target enzymes are lipases

usage examples:

./metarenz cs -i input.fasta -db x.fasta

./metarenz contig_screening input.fasta -db x.fasta -o screenedlipases -lip

./metarenz cs input.fasta -bs 200

>> [ps] or [protein_screening]:

necessary args:

[-i]: Input Protein File (*.fasta)

[-db]: Screening Database File (*.fasta)

optional args:

[-bs]: Bit-Score Filter (Default: 50)

[-o]: Output File Name ==> Default: MeTarEnz_Results/"input_file_name"

[-t]: Number of Threads (Default: 1)

[-lip]: Predict Lipase temperature and pH optima, choose only if your target enzymes are lipases

usage examples:

./metarenz ps -i input.fasta -db x.fasta

./metarenz protein_screening input.fasta -db x.fasta -o screenedlipases -lip

./metarenz ps input.fasta -bs 200

>> [lp] or [lipase_prediction]:

necessary args:

[-i]: Input Protein File (*.fasta)

[-o]: Output File Name

usage examples:

./metarenz lp -i input.fasta -o lipase_pred

./metarenz lipase_prediction -i input.fasta -o lipase_pred

Troubleshoot Common Problems

> If you don't have Java installed, FastQC will not work. install Java (on ubuntu-based systems) using the following commands:

sudo apt update

sudo apt install default-jdk

java --version

> If you faced the error below:

"This sra toolkit installation has not been configured.

Before continuing, please run: vdb-config --interactive

For more information, see https://www.ncbi.nlm.nih.gov/sra/docs/sra-cloud/ "

you can either run the following command:

sratk/vdb-config -i

then press "f" and then press "s".