METAnnotatorX2

This manual dates October 30, 2020

Over the past years, functions and adjustments have been made, which may not be listed here.

Please refer to “-h” or the “parameters” file to explore now additions.

Starting from April 2024, the project has been moved to GitHub in order to track changes and keep the manual updated.

For any suggestion or problem related to METAnnotatorX2: [christian.milani@unipr.it](mailto:christian.milani@unipr.it)

Table of contents

[1. What’s METAnnotatorX2? 3](#_Toc503962913)

[2. What could METAnnotatorX2 do? 3](#_Toc503962914)

[a. Reads taxonomic profiling 3](#_Toc503962915)

[b. Reads functional profiling 3](#_Toc503962916)

[c. Contigs reconstruction and taxonomic profiling 3](#_Toc503962917)

[d. Contigs functional profiling 3](#_Toc503962918)

[3. System requirements 3](#_Toc503962919)

[4. Installation 4](#_Toc503962920)

[5. Software requirements and dependencies 4](#_Toc503962921)

[6. Databases 4](#_Toc503962922)

[7. Input data 5](#_Toc503962923)

[8. Output data 8](#_Toc503962924)

[9. Usage 9](#_Toc503962925)

# **What’s** METAnnotatorX2**?**

METAnnotatorX2 is a bioinformatic platform for reads- and assembly-based analysis of metagenomics datasets. This pipeline allows the perform taxonomic and a range of functional profiling analyses of next-generation shotgun sequencing data. Shotgun metagenomics datasets can also be assembled and the resulting contigs can be taxonomically classified at the genus or species level with the generation of GenBank files with annotated ORFs for each taxon identified. Contigs corresponding to each genus or species profiles can also be functionally characterized.

# What could METAnnotatorX2 do?

METAnnotatorX2 has four program sections:

## Reads taxonomic profiling

Starting from metagenomic raw reads, METAnnotatorX2 performs a taxonomic classification using a set of databases of reference genomes whose taxonomy was checked and corrected to maximize the accuracy of homology-based taxonomic classification of reads.

## Reads functional profiling

Starting from metagenomic raw reads, METAnnotatorX2 performs the reads’ functional classification using the custom Glycobiome and custom Pathways databases.

## Contigs reconstruction and taxonomic profiling

Starting from metagenomic raw reads, METAnnotatorX2 performs the assembly followed by contigs selection, quality controls, ORFs prediction, and gene annotation, concluding with the contigs’ taxonomic classification of the generated GenBank files corresponding to each taxon identified at the genus or species level.

## Contigs functional profiling

GenBank files corresponding to each taxon identified at the genus or species level can be functionally characterized using the custom Glycobiome and custom Pathways databases.

# System requirements

METAnnotatorX2 should run on all Unix platforms, although it has been tested only in different Ubuntu LTS versions, such as v. 16.04, v. 18.04, and v. 20.04. Notably, if METAnnotatorX2 will be executed in a different Unix platform than Ubuntu, the source code of several programs integrated into the pipeline should be downloaded and compiled by the user.

**The user needs to have at least 1TB of free disk space to download all the requested databases.**

# Installation

METAnnotatorX2 is a bash script, so it doesn’t need to be compiled. However, to perform a complete analysis, several extra programs are invoked by METAnnotatorX2. Most of the requested programs are included in the METAnnotatorX2 package, while additional dependencies, listed in the next paragraph, will be installed into the system together with METAnnotatorX2.

**To install METAnnotatorX2, download the installer from the following link and follow the instructions:**

[**http://probiogenomics.unipr.it/sw/METAnnotatorX2/METAnnotatorX2\_installer.sh**](http://probiogenomics.unipr.it/sw/METAnnotatorX2/METAnnotatorX2_installer.sh)

**Then, execute the program as superuser typing “sudo ./METAnnotatorX2\_installer.sh”**

# Software requirements and dependencies

In the METAnnotatorX2 package are included the following programs:

* samtools (Li at al., 209, Bioinformatics)
* prodigal (Hyatt et al., 2010, BMC Bioinformatics.)
* fastq-mcf (https://github.com/ExpressionAnalysis/ea-utils/blob/wiki/FastqMcf.md)
* blast software suite (Morgulis et al., 2008, Bioinformatics)
* RapSearch2 (Ye et al., 2011, BMC Bioinformatics.)
* SPAdes v3.14.0 (Bankevich et al., 2012, J Comput Biol.)
* CANU v2.0 (Koren et al., 2017, Genome Res.)
* Bowtie2 v2.4.0 (Langmead et al., 2012, Nature Methods.)

METAnnotatorX2 requires the following programs or package for full functionality:

* Java version 1.7 or superior
* readseq (type “**sudo apt-get install readseq**” to install)
* gawk (type “**sudo apt-get install gawk**” to install)
* artemis (type “**sudo apt-get install art**” to install)
* EMBOSS software suite (type “**sudo apt-get install emboss**” to install)

**Above listed software will be installed during the installation of MEGAnnotatorX2 using the script “METAnnotatorX2\_installer.sh”. Thus, if you have corrected installed METAnotatorX2, you can ignore the software dependencies above listed.**

# Databases

To perform the reads- and assembly-based analyses, METAnnotatorX2 requires a range of databases to be downloaded: a manually curated RefSeq NCBI (nucleotide) database for the taxonomic classification, a RefSeq NCBI (amino acid) database for functional predictions, a custom Glycobiome database, a custom UniProt database, and multiple host databases for data filtering.

**All these databases will be downloaded during the installation of the METAnnotatorX2 package. Furthermore, databases can be updated by typing “METAnnotatorX2 -u” and following the instructions.**

Besides, during the installation of MEGAnnotatorX2, multiple manually curated RefSeq NCBI databases for the taxonomic classification were provided, i.e., a viral, a prokaryote, and a eukaryote only database. Notably, for the analysis of viromes, we suggest using the viral only database to avoid misclassification due to phage sequences integrated into microorganisms’ chromosomes.

# Input data

METAnnotatorX2 accepts both single-end, paired-end, and long reads data in .fastq format. Then, for the correct execution of the pipeline, refer to the file “parameters” containing a detailed list of analyses and settings.

Here you can find the complete list of the editable parameters implemented in the METAnnotatorX2 pipeline (the reported list reflects the settings order in the parameters file):

Databases:

* Taxonomy database: the path of the RefSeq NCBI database (formatted with makeblastdb).
* Annotation database: the path of the RefSeq NCBI database (formatted with prerapsearch).
* Glycobiome database: the path of the custom glycobiome database (formatted with prerapsearch).
* Pathways database: the path of the custom pathways database (formatted with prerapsearch).
* *Bos taurus* database: the path of the filtering database of *Bos Taurus* (formatted with bowtie2).
* *Canis lupus* database: the path of the filtering database of *Canis lupus* (formatted with bowtie2).
* *Equus caballus* database: the path of the filtering database of *Equus caballus* (formatted with bowtie2).
* *Felis catus* database: the path of the filtering database of *Felis catus* (formatted with bowtie2).
* *Gallus gallus* database: the path of the filtering database of *Gallus gallus* (formatted with makeblastdb).
* *Homo sapiens* database: the path of the filtering database of *Homo sapiens* (formatted with bowtie2).
* *Mus musculus* database: the path of the filtering database of *Mus musculus* (formatted with bowtie2).
* *Rattus norvegicus* database: path of the filtering database of *Rattus norvegicus* (formatted with bowtie2).
* *Sus scrofa* database: the path of the filtering database of *Sus scrofa* (formatted with bowtie2).
* Custom database: /Full\_Path\_of\_Custom\_Database

System:

* Number of threads: number of multiple threads for multithreading programs included in the pipeline (based on the CPU technology used for the analyses) (numerical variable).

Analyses:

* Reads length: length of short reads (numerical variable).
* Reads analyses: set “OK” to perform reads based analyses, otherwise set to “NO”.
* Contigs analyses: set “OK” to perform assembled contigs based analyses, otherwise set to “NO”.

Reads filtering cutoff:

* Reads minimum length: smaller reads will be removed during the filtering step (numerical variable).
* Reads minimum quality value: lower quality reads will be removed during the filtering step (numerical variable).
* *Bos taurus* filtering: set “OK” to perform *Bos taurus* filtering, otherwise set to “NO”.
* *Canis lupus* filtering: set “OK” to perform *Canis lupus* filtering, otherwise set to “NO”.
* *Equus caballus* filtering: set “OK” to perform *Equus caballus* filtering, otherwise set to “NO”.
* *Felis catus* filtering: set “OK” to perform *Felis catus* filtering, otherwise set to “NO”.
* *Gallus gallus* filtering: set “OK” to perform *Gallus gallus* filtering, otherwise set to “NO”.
* *Homo sapiens* filtering: set “OK” to perform *Homo sapiens* filtering, otherwise set to “NO”.
* *Mus musculus* filtering: set “OK” to perform *Mus musculus* filtering, otherwise set to “NO”.
* *Rattus norvegicus* filtering: set “OK” to perform *Rattus norvegicus* filtering, otherwise set to “NO”.
* *Sus scrofa* filtering: set “OK” to perform *Sus scrofa* filtering, otherwise set to “NO”.
* Custom filtering: set “OK” to perform filtering of a custom database, otherwise set to “NO”.

Reads analyses:

* Number of reads: number of reads to be used for reads analyses (numerical variable). Where the number is higher than the number of reads in input, the complete pool of reads will be used for the analyses. Notably, the value must be greater than 100.
* Taxonomy reads e-value cutoff: alignments with higher e-value than the parameter will be discarded during the reads classification versus the RefSeq NCBI database (numerical variable). Example: 5 stands for e-5.
* Glycobiome reads classification: set “OK” to perform glycobiome classification based on input reads, otherwise set to “NO”.
* Pathways reads classification: set “OK” to perform pathways classification based on input reads, otherwise set to “NO”.

Contigs analyses:

* Contigs length: shorter assembled contigs than the parameter will be discarded (numerical variable). Notably, the value must be greater than 200.
* Taxonomy contigs e-value cutoff: alignments with higher e-value than the parameter will be discarded during the contigs classification versus the RefSeq NCBI database (numerical variable). Example: 5 stands for e-5.
* Long reads estimated genome size: set the predicted amount of assembled data by CANU (alphanumeric variable). Example: 30m stands for 30 Megabases.
* Genbank creation at genus level: set “OK” to generate annotated genbank file based on genera classification, otherwise set to “NO”.
* Genbank creation at species level: set “OK” to generate annotated genbank file based on species classification, otherwise set to “NO”.
* Annotation e-value cutoff: alignments with higher e-value than the parameter will be discarded during the annotation of the genes within genbank files (numerical variable). Example: 5 stands for e-5.
* Annotation minimum length: alignments shorter than the parameter will be discarded during the annotation of the genes within genbank files (numerical variable).
* Glycobiome genus classification: set “OK” to perform glycobiome classification based on genes predicted at genus level, otherwise set to “NO”.
* Glycobiome genus e-value cutoff: alignments with higher e-value than the parameter will be discarded during the glycobiome classification of genbank classified at genus level (numerical variable). Example: 5 stands for e-5.
* Glycobiome genus minimum length: alignments shorter than the parameter will be discarded during the glycobiome classification of genbank classified at genus level (numerical variable).
* Pathways genus classification: set “OK” to perform pathways classification based on genes predicted at genus level, otherwise set to “NO”.
* Pathways genus e-value cutoff: alignments with higher e-value than the parameter will be discarded during the pathways classification of genbank classified at genus level (numerical variable). Example: 5 stands for e-5.
* Pathways genus minimum length: alignments shorter than the parameter will be discarded during the pathways classification of genbank classified at genus level (numerical variable).
* Glycobiome species classification: set “OK” to perform glycobiome classification based on genes predicted at species level, otherwise set to “NO”.
* Glycobiome species e-value cutoff: alignments with higher e-value than the parameter will be discarded during the glycobiome classification of genbank classified at species level (numerical variable). Example: 5 stands for e-5.
* Glycobiome species minimum length: alignments shorter than the parameter will be discarded during the glycobiome classification of genbank classified at species level (numerical variable).
* Pathways species classification: set “OK” to perform pathways classification based on genes predicted at species level, otherwise set to “NO”.
* Pathways species e-value cutoff: alignments with higher e-value than the parameter will be discarded during the pathways classification of genbank classified at species level (numerical variable). Example: 5 stands for e-5.
* Pathways species minimum length: alignments shorter than the parameter will be discarded during the pathways classification of genbank classified at species level (numerical variable).

# Output data

Results are distributed in the output folder within sub-folders listed below:

* Assembled\_contigs\_taxonomy

Taxonomic profiling of assembled contigs. “Project\_name\_genera” and “Project\_name\_species” files contain the predicted taxonomy for each contig at genera and species level, respectively.

* Project\_name\_assembly

Assembly of the input reads. The collection of contigs is listed in the fasta file named “contigs.fasta”.

* Filtered\_reads

Filtered reads used in the analysis and statistics.

* Genbank\_genus\_level

Each genbank file contains contigs predicted to belong to a specific genus. Within the GenBank file, the contigs are represented by fasta\_records. For a correct visualization of the GenBank file, we suggest the utilization of the free software Artemis.

* Genbank\_species\_level

Each genbank file contains contigs predicted to belong to a specific species. Within the GenBank file, the contigs are represented by fasta\_records. For a correct visualization of the GenBank file, we suggest the utilization of the free software Artemis.

* Genera\_functional\_prediction

Functional profiling results of glycobiome and pathways based on genes predicted at genera level.

* Reads\_functional\_prediction

Functional profiling results of glycobiome and pathways based on filtered reads.

* Reads\_taxonomy

Taxonomic profiling of filtered reads. In the “Raw\_Output” folder is reported the predicted taxonomy profiles based on the number of hits. In the “Normalized\_Output” folder is reported the predicted taxonomy profiles normalized based on the microorganisms’ genome sizes. The classification is reported at the genus and species level in different files, encompassing the classification of Prokaryotes, Viruses, Fungi and Protists. The file unclassificable\_reads.txt reports the reads with best-hit against different species while the file unclassificable\_reads\_LCA\_classification.txt contains the LCA prediction (if an internet connection isn’t available, LCA prediction will not be performed).

* Species\_functional\_prediction

Functional profiling results of glycobiome and pathways based on genes predicted at species level.

# Usage

METAnnotatorX2 has been placed in the $PATH to be executed as the bash command “**METAnnotatorX2**”.

Using the help arguments, the bash version of METAnnotatorX2 (**METAnnotatorX2 -h**) will display the following message to guide the user in the program usage:

METAnnotatorX2 v1.0 (2020 Oct 30)

usage: METAnnotatorX2 -n [name] -s -i [file] single-end reads mode

or: METAnnotatorX2 -n [name] -p -f [file] -r [file] paired-end reads mode

or: METAnnotatorX2 -n [name] -l -i [file] long reads mode

or: METAnnotatorX2 -n [name] -y -i [file] -f [file] -r [file] hybrid reads mode

Arguments:

-n Project name (alphanumerical variable)

-t Threads number (numerical variable)

-s Single reads technology

-p Paired reads technology

-l Long reads technology

-y Hybrid reads technology

-i Input reads single path (to be used in -s or -l or -y mode)

-f Forward reads path (to be used in -p or -y mode)

-r Reverse reads path (to be used in -p or -y mode)

-h Print Help (this message) and exit

-u Update databases and exit

The bash command can be integrated into a bash script to queue multiple METAnnotatorX2 executions easily. Follow the instruction to use the program with paired-end reads (-p option), single-end reads (-s option), long reads (-l option), or hybrid reads (-y option) as input.

The program can be executed in any location across the machine in which MetannotatorX2 has been installed. MetannotatorX2 will generate hyperlinks of each input file and the software folder. The parameters file will then be copied from the METannotatorX2 folder if no parameters file has been provided in the location.