# DATMA USER MANUAL

**14 de enero de 2020**

Version 3.0

# Contents

# 1. DESCRIPTION

DATMA is a distributed automatic pipeline for fast metagenomic analysis that includes: sequencing quality control, 16S-identification, reads binning, de novo assembly, ORF detection, and taxonomic annotation. DATMA uses a distributed computing model that allows that different stages can be executed in multiple resources reducing the analysis time. DATMA is freely available at: `https://github.com/andvides/DATMA`.

# 2. QUICK START

DATMA is available for the BIOCONDA repository. You should install *conda* before running the following commands.

## 2.1. External tools

To run, DATMA is necessary to install the following tools:
-blast 2.6.0
-bowtie2 2.3.5
-bwa 0.7.17
-checkm-genome 1.0.12
-fastqc 0.11.8
-kraken 1.1
-krona 2.7
-megahit 1.1.3
-pplacer 1.1.alpha19
-quast 5.0.2
-samtools 1.9
-spades 3.13.0
-trimmomatic

These tools can be installed using the following CONDA's command:

$$conda\ install\ <target\ tool>$$

## 2.2. Custom tools

DATMA requires of:
-rapifilt
-selectFasta
-mergeNotCombined
-CLAME
The BIOCONDA file for each tool is available in the DATMA GitHub. To install these tools run:

$$conda\ install\ rapifilt$$
$$conda\ install\ selectFasta$$
$$conda\ install\ mergenotcombined$$

```
conda install CLAME
```

## 2.3. Installation

To install DATMA, download the CONDA package and run the following command.

```
conda install datma
```

## 2.4. Sequential Execution

Execute the runDATMA.sh script use as arguments: the path to the configuration file and enable the sequential mode (seq). Example:

```
$runDATMA.sh configBmini.txt seq
```

Modify the configuration file to set the path to the raw reads and tools.

## 2.5. Distributed Execution

It requires that COMPSs has been installed (see section 5). Use the BIOCONDA repository to install DATMA in all the computers that conform the grid computing. Use the configuration files to administrate the DATMA workflow and enable DATMA in distributed mode (compss). Example:

```
$runDATMA.sh configBmini.txt compss
```

Modify the configuration file to set the path to the raw reads and tools.

## 2.6. Results

DATMA generates the following directories:

- **clean**: Quality filter sequences.
- **16sSeq**: 16S rna Ribosomal sequences detected.
- **bins**: All bins generated by CLAME.
- **clean**: Quality filter results.
- **readsForbin.fastq**: Balance reads that were not binned.
- **round_*_b***: Report directories for the bin stages.
- ***.html**: Report files in HTML format.

# 3. LINUX INSTALL

DATMA is written in Python and has been tested in Linux with Ubuntu distribution. Essential tools and their dependencies have included in an installation script, which requires *sudo* privileges. However, a manual installation could be needed if not all the dependencies were previously installed.

## 3.1. Installation

Execute the *install_datma.sh* script, with sudo properties, to install DATMA (it does not include COMPSs support) and the framework tools.

```
$sudo install_datma.sh
```

## 3.2. Execution

Execute the runDATMA.sh script use as arguments: the path to the configuration file and enable the sequential mode (seq) or distributed (compss), it requires that COMPSs has been installed (see section 5). Example:

```
$runDATMA.sh configBmini.txt seq
```

or

```
$runDATMA.sh configBmini.txt compss
```

Modify the configuration file to set the path to the raw reads and tools.

# 4. MANUAL INSTALLATION

## 4.1. DATMA source codes

Download DATMA source codes into a specific folder. Source codes can be download from: `https://github.com/andvides/DATMA`.

## 4.2. Download Framework tools

### 4.2.1. Quality Trimming and Filtering of Raw Reads tools

- **RAPIFILT**: It can be downloaded from `https://github.com/andvides/RAPIFILT`.
- **Trimmomatic**: It can be downloaded from `http://www.usadellab.org/cms/index.php?page=trimmomatic`.

### 4.2.2. 16S-identification

- **Rfam**: It can be downloaded from `https://rfam.readthedocs.io/en/latest/ftp-help.html`.
- **RDP**: It can be downloaded from `https://rdp.cme.msu.edu/index.jsp`.
- **SILVA** It can be downloaded from `https://www.arb-silva.de/`

### 4.2.3. CLAME binning

- **genFM9**: It can be downloaded from `https://github.com/andvides/CLAME`.
- **mapping**: It can be downloaded from `https://github.com/andvides/CLAME`.
- **binning** It can be downloaded from `https://github.com/andvides/CLAME`

### 4.2.4. Assembler

- **Velvet**: It can be downloaded from `https://www.ebi.ac.uk/~zerbino/velvet/`.
- **Spades**: It can be downloaded from `http://cab.spbu.ru/software/spades/`.
- **MegaHit** It can be downloaded from `https://github.com/voutcn/megahit`
- **Quast** It can be downloaded from `http://bioinf.spbau.ru/quast`

### 4.2.5. Gene detection and Annotation

- **Prodigal**: It can be downloaded from `https://github.com/hyattpd/Prodigal`.
- **GeneMark**: It can be downloaded from `https://ngs.csr.uky.edu/GeneMark`.
- **CheckM**: It can be downloaded from `https://ecogenomics.github.io/CheckM/`.
- **BLAST**: It can be downloaded from `https://www.ncbi.nlm.nih.gov/guide/howto/run-blast-local/`
- **KAIJU** It can be downloaded from `http://kaiju.binf.ku.dk/`

### 4.2.6. Final report

- **Krona**: It can be downloaded from `https://github.com/marbl/Krona`

## 4.3. Install Framework tools

Install all the tools according to the author instruction and recommendations and add each one to the execution *PATH*. Probe that each tool can be executed from the DATMA path.

$$\boxed{\$export\ PATH = "/home/\$USER/\$Tool/bin:\$PATH"}$$

# 5. DISTRIBUTED MODE

## 5.1. COMPSs

It can be downloaded from: `https://www.bsc.es/research-anddevelopment/software-and-apps software-list/comp-superscalar`

## 5.2. Wokers configuration

Replicate the DATMA installation process for all the workers.

## 5.3. Database configuration

When the distributed mode is selected, DATMA requires that the database_nt and database_kaiju (see the configuration file.txt) have identical locations in all the workers. If you are using different paths, you can provide a directory with the same path as the master directory and generate symbolic links of each database to this path.

# 6. CONFIGURATION FILES

## 6.1. DATMA configuration file

In this file, the user specifies the input-sequences file, the output directory, the workflow stages, the database directories, the number of threads to use, CLAME's bases parameter, etc.

```
#############################################################################
#############################################################################
##DATMA CONFIGURATION FILE
##
##USE THIS FILE TO PASS THE ARGUMENTS TO DATMA WORKFLOW
##SEE DATMA MANUAL FOR DETAILS
##VERSION 3 01-10-2019
#############################################################################
#
#Stages
#1.Clean de reads
#2.Remove 16S sequences
#3.CLAME rounds
#4.Assemble and Annotate each bin by separated
#5.Final report
#use start_in flag to start in a particular stage (INCLUDE the stage,default 1)
#use end_in flag to start in a particular stage (NOT include the stage, default 6)
#
#############################################################################
#GENERAL PARAMETERS
#############################################################################
-start_in 3
#-end_in 3
-inputFile /home/users/jfar/metagenomaICP/Limpios/Datma/clean_20M_aAI_1.fastq.gz,/home
-outputDir /home/users/jfar/metagenomaICP/Limpios/Datma/20M_aAI
-cpus 30
-typeReads illumina
```

```
#############################################################################
#Quality Control Configuration
#############################################################################
#-cleanTool trimmomatic
-lq 30
-rq 30
-m  60
-wq: 2
-trimmomatic_path /home/software/src/Trimmomatic-0.38/
#############################################################################
#Merge illunima files using FLASH tool
#############################################################################
#fb: Number of bases for set a merge (default 0)
#-fb 5
-forceMerge
#############################################################################
#16S-REMOVE
#############################################################################
-database_16s_fasta  /home/db/16sDatabase/rfam/RFAM_db.fasta
-database_16s_fm9  /home/db/16sDatabase/rfam/rfam.fm9
-RDP_path /home/software/src/RDPTools/
-useBWA
#############################################################################
#CLAME
#############################################################################
-bases 70,50
-sizeBin 5000
-ld 10
-block 20000000
-fm9 /home/users/jfar/metagenomaICP/Limpios/Datma/20M_aAI/readsForbin
#use -aux_clase to pass other parameter example tol or fw bases
-clame_aux -tol 0.5
#############################################################################
#ASSEMBLY OPTIONS
#############################################################################
```

```
-assembly spades
-asm_aux --only-assembler
#############################################################################
#ASSEMBLY QUALITY WITH QUAST TOOL OPTIONS
#############################################################################
#ref_name="PATH to ref used to asses the bin completeness"
#-use_ref
#-ORF_aux -p meta
#
#############################################################################
#BLAST
#############################################################################
-database_nt  /home/db/NT/NT_May_2019/nt
#############################################################################
#Kaiju
#############################################################################
-database_kaiju  /home/software/kaiju/kaijudb/
#############################################################################
#Krona
#############################################################################
#-combine
```

## 6.2. COMPSs configuration files

- **resources.xml**: provides information about the available execution resources. (see the COMPSs_Installation_Manual file)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ResourcesList>
 <ComputeNode Name="172.16.7.105">
 <Processor Name="master">
 <ComputingUnits>1</ComputingUnits>
 </Processor>
 <Adaptors>
 <Adaptor Name="integratedtoolkit.nio.master.NIOAdaptor">
 <SubmissionSystem>
```

```
<Interactive/>
</SubmissionSystem>
<Ports>
<MinPort>40001</MinPort>
<MaxPort>43002</MaxPort>
</Ports>
</Adaptor>
<Adaptor Name="integratedtoolkit.gat.master.GATAdaptor">
<SubmissionSystem>
<Batch>
<Queue>sequential</Queue>
</Batch>
<Interactive/>
</SubmissionSystem>
<BrokerAdaptor>sshtrilead</BrokerAdaptor>
</Adaptor>
</Adaptors>
</ComputeNode>
<ComputeNode Name="172.16.7.104">
<Processor Name="worker">
<ComputingUnits>1</ComputingUnits>
</Processor>
<Adaptors>
<Adaptor Name="integratedtoolkit.nio.master.NIOAdaptor">
<SubmissionSystem>
<Interactive/>
</SubmissionSystem>
<Ports>
<MinPort>40001</MinPort>
<MaxPort>43002</MaxPort>
</Ports>
</Adaptor>
<Adaptor Name="integratedtoolkit.gat.master.GATAdaptor">
<SubmissionSystem>
<Batch>
```

```
 <Queue>sequential</Queue>
 </Batch>
 <Interactive/>
 </SubmissionSystem>
 <BrokerAdaptor>sshtrilead</BrokerAdaptor>
 </Adaptor>
 </Adaptors>
 </ComputeNode>
</ResourcesList>
```

- **project.xml**: provides information about the resources used in a specific execution. (see the COMPSs_Installation_Manual file)

```
<Project>
 <MasterNode/>
 <ComputeNode Name="172.16.7.105">
 <InstallDir>/opt/COMPSs/</InstallDir>
 <WorkingDir>/tmp/COMPSsWorker/</WorkingDir>
 <User>datma_user</User>
 <LimitOfTasks>1</LimitOfTasks>
 <Application>
 <AppDir>/DATMA/codes/</AppDir>
 <Pythonpath>/ DATMA/codes/</Pythonpath>
 </Application>
 </ComputeNode>
 <ComputeNode Name="172.16.7.104">
 <InstallDir>/opt/COMPSs/</InstallDir>
 <WorkingDir>/tmp/COMPSsWorker/</WorkingDir>
 <User>andresb</User>
 <LimitOfTasks>1</LimitOfTasks>
 <Application>
 <AppDir>/DATMA /codes/</AppDir>
 <Pythonpath>/DATMA /codes/</Pythonpath>
 </Application>
 </ComputeNode>
</Project>
```

# 7. RUNNING

## 7.1. Database

Download the 16S RNA database and generate database index. Execute the $build16S_database.sh$ script.
Example:

$$_16S_dataBase.sh\ databasebwa$$

It is necessary only the first time that you install DATMA.

## 7.2. Automatic mode

To easy the execution, DATMA provides a run scripts to execute the complete workflow. Just type the $runDATMA.sh$ script use as arguments: the path to the configuration file and the running mode sequential (seq) or distributed (compss).

$$\$runDATMA.sh\ configBmini.txt\ seq$$

or

$$\$runDATMA.sh\ configBmini.txt\ compss$$

## 7.3. Sequential mode

DATMA can be executed using python command.

$$\$pythondatma\_seq.py\ configBmini.txt\ seq$$

## 7.4. Distributed mode

DATMA can be executed using the runcompss script of COMPSs.

$$\$runcompss\ --project=<project.xml>--resources=<resources.xml>--summary\ -d--lang=pythondatma.py-f<configurationFile.txt>$$

# 8. FAQ

Please contact us:

bernardo.benavides@udea.edu.co
felipe.cabarcas@udea.edu.co