# aureme Documentation

Meziane AITE

Jun 20, 2019

# Contents

**1 Contents 1**

1.1 How to use Docker? . . . 1

1.2 How to use the AuReMe workspace (default workflow) . . . 2

1.3 FAQ . . . 14

1.4 Indices and tables . . . 33

## CHAPTER 1

## Contents

## 1.1 How to use Docker?

**Requirements** Docker (v 1.10 at least)

To install Docker, please follow the instructions on docker.com, considering your operating system*.

- On Mac OS: requires at least Yosemite 10.10.4
- On Windows: requires at least Windows 8

### 1.1.1 Running a Docker container

1. Launch the Docker machine (see the instruction on docker.com). For example:
   - On Fedora: **sudo systemctl start docker**
   - On Mac OS and Windows: run the Docker launcher
2. Download the AuReMe Docker image

```
shell> docker pull dyliss/aureme-img
```

3. To verify that the image has been downloaded correctly, check it in the list of your local images:

```
shell> docker images -a

REPOSITORY                         TAG      IMAGE ID       CREATED      SIZE
docker.io/dyliss/dyliss/aureme-img  latest   6cf38ab4edc8   1 hour ago   1.68 GB
```

4. Create a folder that will serve as a bridge to share data from/to the Docker container. Let us call it **bridge** for instance.

5. Create a Docker container from the following image with this command:

```
shell> docker run -ti -v /my/path/to/the/directory:/shared --name="aureme_cont"␣
↪dyliss/aureme-img bash
```

The path given for -v is the one to the shared directory on your host machine. **This path has to end on the directory name** (without any **/** at the end). **The path has to be complete** (from **/users** or from **C:\\** for Windows users). After the ‘:’ is the name of the mirror directory in the Docker container. Please do not change it.

For Windows users, be careful, you have to indicate your path this way:

You have just made a bridge between **\my\path\to\the\directory\brigde** and the container **aureme_cont**.

You can create as many containers as you wish, as long as you give them different names.

Your AuReMe container is now running and correctly installed. Congratulations! You are now inside the container **aureme_cont**.

### 1.1.2 Some tips about Docker

- To exit the container, tape **exit.**

```
aureme> exit
```

- To list all your containers:

```
shell> docker ps -a

CONTAINER ID   IMAGE                          COMMAND     CREATED        STATUS     ␣
↪PORTS   NAMES          SIZE
fc969ed0d2c7   docker.io/dyliss/aureme-img   "bash"      2 hours ago    Up 5 hours ␣
↪        aureme_cont   11 MB (virtual 3.52 GB)
```

Remark that you can see, with this command, the state of your containers in the STATUS column: **up** (running, you can connect to it), **exited** (stopped, need to be started again).

- To start or stop the container (from your host):

```
shell> docker start aureme_cont
shell> docker stop aureme_cont
```

- If you want to go inside **a started/running** container:

```
shell> docker exec -it aureme_cont bash
```

- To delete a container: **docker rm container_id** ( or **name**)
- To delete an image: **docker rmi image_id** (or **name**)

Before deleting a Docker image, you have to delete all the Docker containers which are linked with the image you would like to remove. And before removing a Docker container, you have to stop it.

AuReMe documentation

## 1.2 How to use the AuReMe workspace (default workflow)

AuReme is deployed in a Docker image. Thanks to this Docker image, all the tools inside the AuReMe workspace are ready to use inside the AuReMe container.

### 1.2.1 Requirements

1. Create your Docker container as explained in the previous step *Running a Docker container*, start the container and go inside.

#### Start a new study

2. Use the following command to start a new study. Choose an identifier for this study (ex: replace **test** by your organism name). In order to illustrate this documentation, we will use **test** as a run identifier.

   ```
   aureme> aureme --init=test
   ```

   Now you will find on your own computer (host), in your **bridge** directory, a folder **test** with many subdirectory and files. This is your work directory, on which AuReMe is going to run.

   **Note:** Notice that from now until the end of the process, every command will be stored as a log in the **bridge > test > log.txt** file. The whole output of these commands will also be stored in the **bridge > test > full_log.txt** file.

   If you wish NOT to store such logs, you can use the **quiet** argument in your command(s). This will redirect the output on the terminal. For example:

   ```
   aureme> aureme --run=test --cmd="some_command" -q
   ```

   For further details on the log files, please see the *How to manage the log files?* chapter.

3. To get an overview of AuReMe, you can get a sample by using this command.

   ```
   aureme> aureme --run=test --sample
   ```

#### Define the reference database

4. The final step is to define which reference database to use. The available databases are listed in your terminal when you create a new study. If needed, use this command to display them again.

   ```
   aureme> aureme --run=test --cmd="getdb" -q
   Available database in Aureme:
   /home/data/database/BIGG/bigg
   /home/data/database/BIOCYC/METACYC/20.5/metacyc_20.5_enhanced
   /home/data/database/BIOCYC/METACYC/22.0/metacyc_22.0_enhanced
   /home/data/database/MODELSEED/modelSeed
   ```

   **This reference database is needed to:**

   - Be able to match all the identifiers of the entities of metabolic networks.
   - Gap-fill the metabolic network in the gap-filling step.
   - Uniforms the data in one unique database.

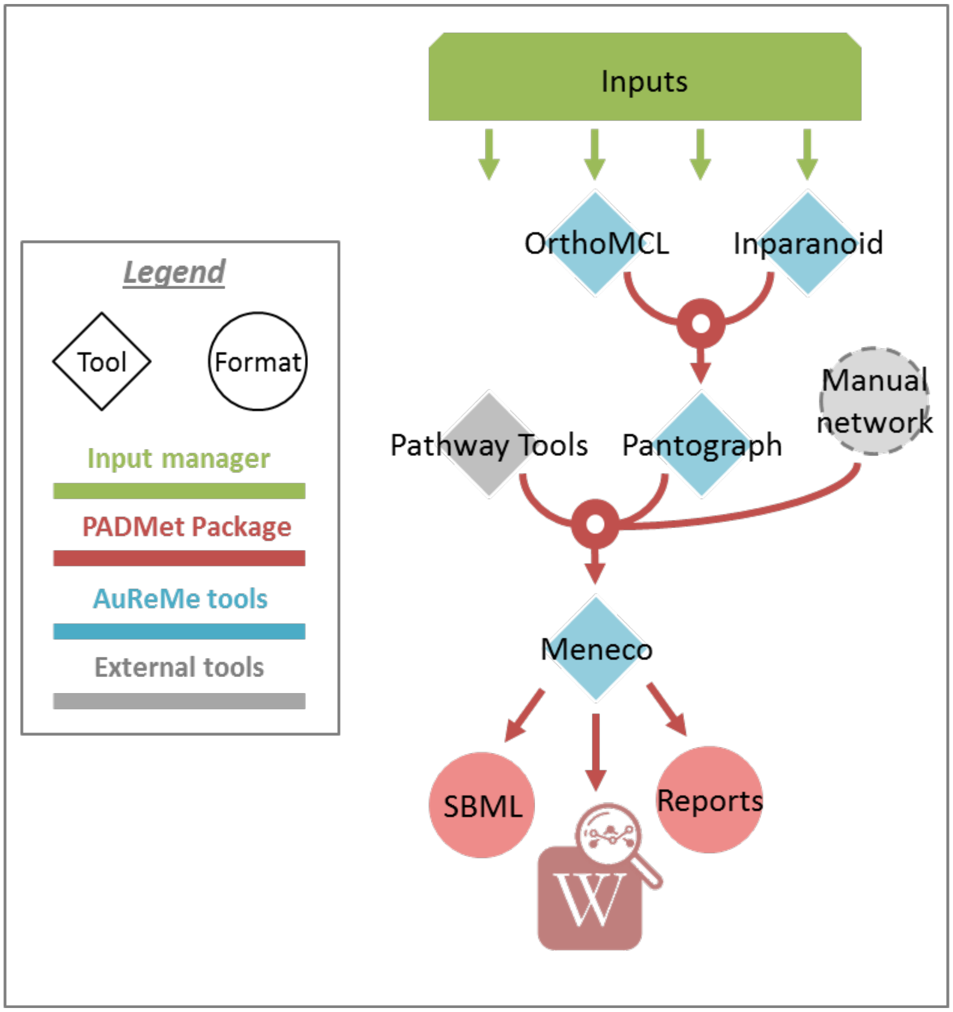
   To select one of the above databases, replace the corresponding path in the configuration file: **config.txt**, in the **DATA_BASE** variable, or comment the line if you don't want/can't use a database.

   The **config.txt** file is stored at the root of your **test** folder.

### 1.2.2 The default workflow

By default, the AuReMe workspace includes an automatic workflow for metabolic network reconstruction. This workflow runs several pre-installed tools and generates diverse output files. The process can be either run entirely in a single command, or run step by step to personalize it or do some intermediary analysis.

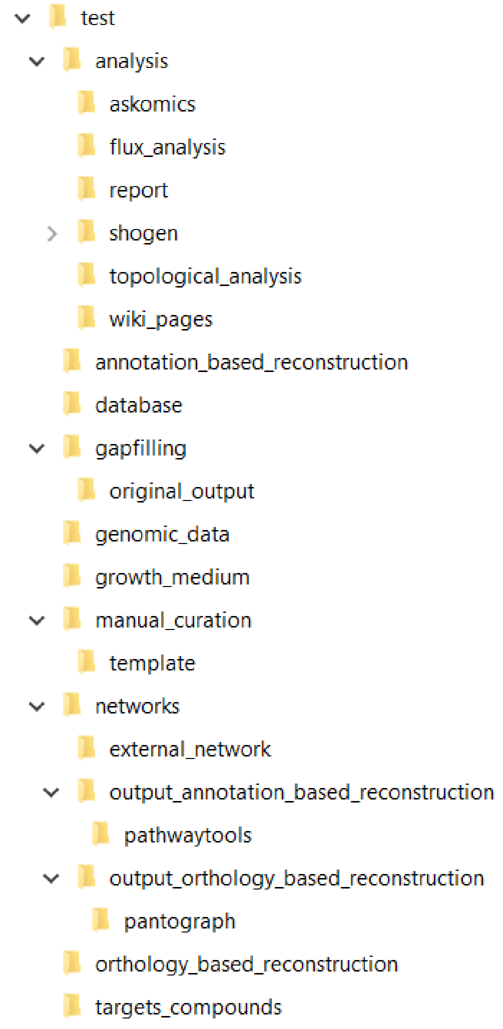
For instance, if you run the **draft** command (see *Merge metabolic networks*), it will run all the previous steps automatically as described in the following figure. This figure details the steps of the default workflow.



### 1.2.3 Data organization

#### Bridge structure

The **bridge** directory will store all your input data you will provide, and all the result files the workflow is going to create. In this section, all the **bridge** sub-directories will be described.



```
test
  analysis
    askomics
    flux_analysis
    report
    shogen
    topological_analysis
    wiki_pages
  annotation_based_reconstruction
  database
  gapfilling
    original_output
  genomic_data
  growth_medium
  manual_curation
    template
  networks
    external_network
    output_annotation_based_reconstruction
      pathwaytools
    output_orthology_based_reconstruction
      pantograph
  orthology_based_reconstruction
  targets_compounds
```

**analysis**: All output files of the analysis processes.

**annotation_based_reconstruction**: If you want to use annotated genomes (to run the annotation-based reconstruction part of the workflow), put here all the output files of the annotation tool. For instance with Tools, copy-paste the whole PGDB *Annotation-based reconstruction* for more details).

**database**: If you want to use your own database put in this folder your database in padmet format, if you have a sbml convert this file to padmet (see *How to convert files to different formats?*). Don't forget to update the **config.txt** file after transforming your database into the padmet format.

**gapfilling/original_output**: If you run the metabolic network reconstruction with gap-filling, will contain all the output files of gap-filling tools or any post-process from AuReMe (see the *Gap-filling* section).

**genomic-data**: The directory in which to put the data from the **orthology_based_ reconstruction** directory (if you have run this part of the workflow).

**orthology_based_reconstruction**: If you want to use model organisms (to run orthology-based reconstruction part of the workflow), put here the proteome (**FAA_model.faa** or **GBK_model.gbk**) and the metabolic network (**metabolic_model.sbml**) of your model (see below *Orthology-based reconstruction* for more details).

**targets_compounds**: Description of the set target compounds (**targets.txt**), that is to say metabolites whose production is supposed to be achieved by the metabolism of the species under study (components of the biomass reaction or orther metabolites), see the

### Provide input files

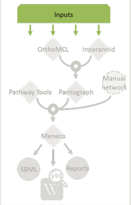
> First of all, you have to provide to AuReMe all the input files needed for the different steps you want to run in the workflow. The steps can be optional or run several times, at different phases of the process. However, you have to store the input data for each steps, observing the architecture described above for the **bridge** directory (see *Bridge structure* section).

Here is the list of inputs you have to provide to run the pre-set default workflow. If you want to run only part of it, please see the corresponding sections and the chapter *How to create a new 'à-la-carte' workflow?*

- See *Orthology-based inputs*.
- See *Annotation-based inputs*.
- **External source for reconstruction** If you already have one or several external metabolic networks for your studied species and you want to improve them, just copy-paste them (SBML format) in the **networks > external_network** folder.

```
/test
  |-- networks
      |-- external_network
          |-- first_manual_network.sbml
          |-- second_manual_network.sbml
          |-- ...
```

### Check input files validity

| | |
|---|---|
| This will verify the format and consistency of your data for a better quality result. Moreover, it will generate all the supplementary files needed by the workflow tools and put them into the corresponding folders. For more information about input files validity see *What is checked in my input files?* |  Inputs<br>OrthoMCL<br>Inparanoid<br>Pathway Tools<br>Pantograph<br>Manual network<br>Meneco<br>SBML<br>Reports |

For this purpose, use this command:

```
aureme> aureme --run=test --cmd="check_input"
```

> **Warning:** Always check the validity of the inputs before running any workflow task, and after having put every input files needed for the steps of the workflow.

### 1.2.4 Orthology-based reconstruction

#### Method: Pantograph

**Input files**

Required for the orthology-based reconstruction (method:

Pantograph):

* Genbank or Proteome of your studied organism (.gbk or .faa)

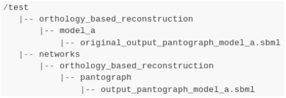
* Genbank or Proteome of your reference organism (.gbk or .faa)

* Metabolic network of your reference organism (.sbml)

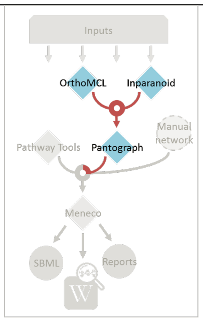
* (option) a dictionary file if genes ids used in metabolic

network are different with gbk/faa (.txt)

**Result files**



```
/test
   |-- orthology_based_reconstruction
       |-- model_a
           |-- original_output_pantograph_model_a.sbml
   |-- networks
       |-- orthology_based_reconstruction
           |-- pantograph
               |-- output_pantograph_model_a.sbml
```



#### Orthology-based inputs

1. Put all the available genomic data of the studied organism in the folder **genomic_data**, either a Genbank (.gbk) or a Fasta (.faa) file.

> **Warning:** Give them these exact names (respectively): GBK_study.gbk and FAA_study.faa.

2. For each reference organism you want to use, create a subdirectory in the directory **orthology_based_reconstruction**. Give it the name of your model organism (e.g. **model_a**). On a Linux operating system, here is the above command to create a new folder named **model_a**.

```
shell> mkdir orthology_based_reconstruction/model_a
```

3. In each folder, put:

- the Genbank file of your model organism, **with the exact name GBK_model.gbk** OR the proteome of your model organism, **with the exact name FAA_model.faa**,
- the metabolic network of your model organism, **with the exact name metabolic_model.sbml**

```
/test
  |-- orthology_based_reconstruction
       |-- model_a (you can change the name of the folder)
            |-- GBK_model.gbk or FAA_model.faa
            |-- metabolic_model.sbml
            |-- dict_genes.txt (option)
```

4. The genome (or proteome) and the metabolic network of your model organism have to contain the same kind genes (or proteins) identifiers to be comparable. If not enough genes (or proteins) are in common between the two files, the process will stop to avoid poor quality data production.

   If you want to pursue on the process, please provide a dictionary file between the gene (or protein) identifiers of these two files. Name this dictionary **dict_genes.txt**. Here is the dictionary file format asked (tabulation separated values):

```
gene_id_from_sbml1\tgene_id_from_faaA
gene_id_from_sbml2\tgene_id_from_faaB
gene_id_from_sbml3\tgene_id_from_faaC
```

### Orthology-based run

**Warning:** Remember to check the validity of the inputs before running any workflow task.

5. If you want to run only the orthology-based reconstruction, use now this command:

```
aureme> aureme --run=test --cmd="check_input"
```

6. To run **only** the orthology-based reconstruction, use this command:

```
aureme> aureme --run=test --cmd="orthology_based"
```

7. Use this command, to get the database of a given metabolic network:

```
aureme> aureme --run=test --cmd="which_db SBML=output_pantograph_model_a.sbml"
```

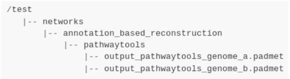
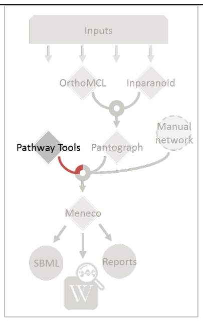
**Warning:** Because the metabolic network from the reference organism could came from different databases, it's critical to check the database of each network and if needed convert the network to your reference database selected (see *How to use the AuReMe workspace (default workflow)* and *Define the reference database*).

The previous command will check the database of the file output_pantograph_mode_a.sbml, if the database is different for the reference, use the next command to create a mapping file to the reference database. For more information about sbml mapping see *How to map a metabolic network on another database?*

```
aureme> aureme --run=test --cmd="sbml_mapping SBML=output_pantograph_model_a.sbml␣
↪DB=METACYC"
```

### 1.2.5 Annotation-based reconstruction

#### Method: Pathway Tools

| | |
|---|---|
| **Input files**<br>Required for the annotation-based reconstruction (method:<br>Pathway Tools):<br>* The output of Pathway tools (PGDB folder)<br>**Result files**<br><br>/test<br>    \|-- networks<br>        \|-- annotation_based_reconstruction<br>            \|-- pathwaytools<br>                \|-- output_pathwaytools_genome_a.padmet<br>                \|-- output_pathwaytools_genome_b.padmet |  Inputs<br>OrthoMCL<br>Inparanoid<br>Manual network<br>Pathway Tools<br>Pantograph<br>Meneco<br>SBML<br>Reports |

#### Annotation-based inputs

1. Put the output of Pathway Tools (the whole PGDB directory) in the folder **annotation_based_reconstruction**.

```
/test
  |-- annotation_based_reconstruction
      |-- genome_a (you can change the name of the folder)
          |-- compounds.dat
          |-- enzrxns.dat
          |-- genes.dat
          |-- pathways.dat
          |-- proteins.dat
          |-- reactions.dat
```

2. The above cited files are required in order to run the Annotation-based reconstruction. If you have run several times Pathway Tools and want to use all of these annotations, just copy-paste the other PGDB folders in the **annotation_based_reconstruction** directory.

### Annotation-based run

3. If you want to run only the annotation-based reconstruction, use now this command:

```
aureme> aureme --run=test --cmd="check_input"
```

> **Warning:** Remember to check the validity of the inputs before running any workflow task.

4. To run **only** the annotation-based reconstruction, use this command.

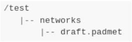
```
aureme> aureme --run=test --cmd="annotation_based"
```

## 1.2.6 Merge metabolic networks

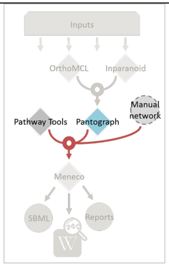
**Input files**

Metabolic networks in the **networks** directory.

**Result file**



```
/test
   |-- networks
       |-- draft.padmet
```



To merge all available networks from the **networks** directory into one metabolic network, merging all data on the studied species, run this command:

```
aureme> aureme --run=test --cmd="draft"
```

> **Note:** You can also add external metabolic network to create the draft (see *Data organization*).

> **Warning:** Before merging your networks, check if not already done if all the SBML are using the reference database. Also check the compartment ids used in each of them, delete and change compartment if need.

For example: if a SBML is using KEGG database but your reference database is Metacyc, you will have to map this SBML to create a mapping file which will be used automatically in the merging process.

If a SBML contains a compartment id like 'C_c' and another contains 'c', although they correspond to the same compartment 'cytosol' because of different ids, a compound in 'C_c' is not the same as a compound in 'c', therefore there will be a loss of connectivity in the network. see *How to map a metabolic network on another database?* and *How to manage metabolic network compartment?*

## 1.2.7 Gap-filling

### Method: Meneco

**Input files**

Required for the gap-filling (method: Meneco):

* A metabolic network reference database (.padmet or .sbml)

(Metacyc 20.5, 22.0, BIGG and ModelSeed are available by

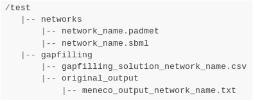
default)

* Seed and target metabolites (.txt)

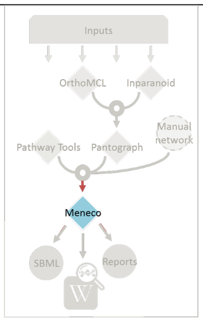
* A metabolic network to fill (typically created during the

previous steps).

**Result files**



```
/test
   |-- networks
       |-- network_name.padmet
       |-- network_name.sbml
   |-- gapfilling
       |-- gapfilling_solution_network_name.csv
       |-- original_output
           |-- meneco_output_network_name.txt
```



### Gap-filling input

1. You must have selected a reference database to fill-in the potential gaps in the metabolic network. If it is not done yet, please see *Define the reference database*.
2. Put the seeds file (named **seeds.txt**) in the **growth_medium** folder. The seed compounds are the description of the set of metabolites that is available to initiate the metabolism (growth medium). Put also the artefacs file (named **artefacts.txt**) in the growth_medium folder. The artefacts file has the same format as the seeds file. Here, artefacts are metabolites allow Meneco to initiate cycles in in a metabolic network (report to *What are "artefacts"?* section). Here is as example of the seed file format:

```
seed_name_compound_id1\tcompartment1
seed_name_compound_id2\tcompartment2
seed_name_compound_id3\tcompartment3
```

3. Set the growth medium using this command:

```
aureme> aureme --run=test --cmd="set_medium NETWORK=network_name NEW_NETWORK=new_
↪network_name"
```

For more details on the medium settings, see *How to manage growth medium?*

**Warning:** If you don't precise any **NEW_NETWORK** name, the current network will be overwritten.

4. Put the target file (named **targets.txt**) in the **targets_compounds** folder. The targets are metabolites whose production is supposed to be achieved by the metabolism of the species under study (components of the biomass reactions or other metabolites). Here is as example of the target file format:

> target_name_compound_id1\tcompartment1
> target_name_compound_id2\tcompartment2
> target_name_compound_id3\tcompartment3

5. You will have to indicate which metabolic network you want to gap-fill with the Meneco software. If you want to gap-fill a network created in the previous steps, there is nothing to do. Otherwise, put the network you want to gap-fill (PADMET format) in the **networks** directory.

```
/test
  |-- networks
      |-- network_name.padmet
  |-- growth_medium
      |-- seeds.txt
      |-- artefacts.txt
  |-- targets_compounds
      |-- targets.txt
```

### Gap-filling run

6. (optional step) To generate the gap-filling solution run this command:

```
aureme> aureme --run=test --cmd="gap_filling_solution NETWORK=network_name"
```

**Note:** Do not forget the quotation marks.

It will calculate the gap-filling solution on the **network_name** network (in the **networks** directory) and put it into the **gapfilling** directory as **gapfilling_solution_network_name.csv**.

7. To generate the gap-filled network (and run step 6), run this command:

```
aureme> aureme --run=test --cmd="gap-filling NETWORK=network_name NEW_NETWORK=new_
↪network_name"
```

**Note:** Do not forget the quotation marks.

It will calculate the gap-filling solution (if it is not yet done) on the **network_name** network (in the **networks** directory) and put it into the **gapfilling** directory. Then it will generate the metabolic network (**new_network_name**), completed with the gap-filling solution, in the **networks** directory.

**Note:** You can first generate the solution, modify it, then generate the gap-filled network.

**Warning:** If you don't precise any **NEW_NETWORK** name, the current network will be overwritten.

### 1.2.8 Manual curation

This step can be done several times and at any moment of the workflow.

1. Describe the manual curation(s) you want to apply by filling the corresponding form(s) as explained below.

**Warning:** It is highly recommanded to create a new form file (.csv) each time you want to apply other changes, in order to keep tracks of them.

#### Add a reaction from the database or delete a reaction in a network

a) Copy from the folder **manual_curation > template** the file **reaction_to_add_delete.csv** and paste it into the **manual_curation** directory (this way on Linux operating systems):

```
aureme> cp manual_curation/template/reaction_to_add_delete.csv manual_curation/my_
↪create_form.csv
```

b) Fill this file (follow the exemple in the template).

```
idRef        Comment                                 Action   Genes
my_rxn       Reaction deleted because of x reason    delete
RXN-12204    Reaction added because of x reason      add      (gene1 or gene2)
RXN-12213    Reaction added because of x reason      add      gene18
RXN-12224    Reaction added because of x reason      add
```

#### Create new reaction(s) to add in a network

a) Copy from the folder **manual_curation > template** the file **reaction_creator.csv** and paste it into the **manual_curation** directory (this way on Linux operating systems):

```
aureme> cp manual_curation/template/reaction_creator.csv manual_curation/my_
↪create_form.csv
```

b) Fill this file (follow the exemple in the template).

```
reaction_id   my_rxn
comment       reaction added because of X reason
reversible    false
linked_gene   (gene_a or gene_b) and gene_c
#reactant/product   #stoichio:compound_id:compart
```

(continues on next page)

(continued from previous page)

```
reactant            1.0:compound_a:c
reactant            2.0:compound_b:c
product             1.0:compound_c:c

reaction_id   my_rxn_2
comment       reaction added because of X reason
reversible    true
linked_gene
#reactant/product   #stoichio:compound_id:compart
reactant            1.0:compound_a:c
reactant            2.0:compound_d:c
product             1.0:compound_c:c
product             1.0:compound_d:c
```

### Apply changes

2. To apply the changes described in the **my_form_file.csv** form file, run this command:

```
aureme> aureme --run=test --cmd="curation NETWORK=network_name NEW_NETWORK=new_
↪network_name DATA=my_form_file.csv"
```

**Warning:** If you don't precise any **NEW_NETWORK** name, the current network will be overwritten.

## 1.3 FAQ

### 1.3.1 Can I have a sample of AuReMe?

To get an overview of AuReMe, you can get a sample by using this command:

```
aureme> aureme --run=test --sample
```

You will get a folder named 'aureme_sample' in your bridge directory (i.e: /home/data/run_template/aureme_sample). This folder contains all input and o utput files as if you had run the entire metabolic network reconstruction workflow for the example files about *Tisochrysis lutea* (microalgae). Look at the logs file to understand the different commands used in the reconstruction process.

**Note:** if you do not want to pollute your log files when testing things in your sample run, do not forget to use the **quiet (-q)** argument in your command(s) if you wish **not** to store any log, this way:

```
aureme> aureme --run=aureme_sample --cmd="cmd args" -q
```

### 1.3.2 How to convert files to different formats?

The AuReMe workspace natively provides several functions for formats conversion, through the PADMet Python package. The available convertors are:

- From sbml to padmet format:

```
aureme> aureme --run=test --cmd="draft"
```

This command will convert all sbml in networks folder of '**test**' to one padmet. If you want to convert one sbml to padmet format, simply put this file in networks folder of your run and make sure there is no other sbml file nor padmet file, either in networks directory, or in one of the sub-directory of networks. Then run the command.

If you want to merge many sbml to one padmet, add all of them in **networks > external_network** folder then run the command. Ensure that there is no other sbml nor padmet file, either in networks directory, or in one of the sub-directory of networks before running the command. In the case one sbml would be forgotten, it could add to the resulted **draft.padmet** or another reading error could occur.

- From padmet to sbml format:

```
aureme> aureme --run=test --cmd="padmet_to_sbml NETWORK=my_network [LVL=3]"
```

This command will convert the padmet file **my_network.padmet** from networks folder of '**test**' to create a sbml file **my_network.sbml**. By default the sbml level is set to '**3**', you can change the default value in the config.txt file or with the argment LVL (3 or 2)

- From txt to sbml format:

```
aureme> aureme --run=test --cmd="compounds_to_sbml CPD=/path/to/file/root_txt_file
↪"
```

This command will convert a txt file containing compounds ids to a sbml file **/path/to/txt_file.sbml**. The txt file must contain one compound id by line and optionally the compartment of the id which by default is 'c'. Example of file:

```
ATP
ADP
WATER\tC-BOUNDARY
LIGHT\tC-BOUNDARY
```

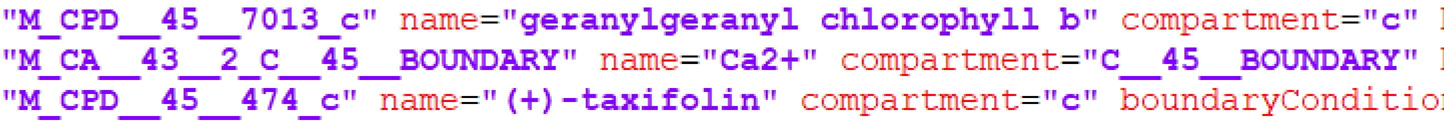
- From GFF/GBK to FAA format:

**Note:** AuReMe integrate some scripts from padmet-utils tools, for example, gbk_to_faa command use the script /programs/padmet-utils/connection/gbk_to_faa.py. Not all functions are encapsulated in AuReMe, there is a lot of scripts that could be helpful. For more information, see https://github.com/AuReMe/padmet-utils.

```
aureme> aureme --run=test --cmd="gbk_to_faa GBK_FILE=/path/to/gbk_file OUTPUT=/path/
↪to/output_file"
```

### 1.3.3 How to manage growth medium?

In AuReMe, a compound is defined as a part of the growth medium (or 'seeds' for gap-filling tools) if this compound is in the compartment 'C-BOUNDARY'.



```
"M_CPD__45__7013_c" name="geranylgeranyl chlorophyll b" compartment="c"
"M_CA__43__2_C__45__BOUNDARY" name="Ca2+" compartment="C__45__BOUNDARY"
"M_CPD__45__474_c" name="(+)-taxifolin" compartment="c" boundaryConditio
```

The growth medium is linked to the metabolic network by two reactions, a non-reversible reaction named ‘*TransportSeed-compound-id*’ which transport a compound of the growth medium from the compartment ‘C-BOUNDARY’ to the ‘e’ (extra-cellular) and a reversible reaction named ‘*ExchangeSeed-compound-id*’ which exchange the same compound from ‘e’ to the ‘c’ (cytosol). When creating a sbml file, the compounds in the ‘C-BOUNDARY’ compartment will be set as ‘BOUNDARY-CONDITION=TRUE’ to allow flux (see http://sbml.org/Documents/FAQ#What_is_this_.22boundary_condition.22_business.3F).

**Note:** Some metabolic networks manage the growth medium with a reversible reaction which consume nothing and produce a compound in the ‘c’ compartment. We chose not to do the same for clarity and because this metod made crash some dedicated tools for metabolic network .

- Get the list of compounds corresponding to the growth medium of a network in padmet format:

  ```
  aureme> aureme --run=test --cmd="get_medium NETWORK=network_name"
  ```

  Return a list of compounds or an empty list

- Set the growth medium of a network in padmet format:

  ```
  aureme> aureme --run=test --cmd="set_medium NETWORK=network_name [NEW_NETWORK=new_
  ↪network_name]"
  ```

This command will remove the current growth medium if existing, then create the new growth medium by adding the required reactions as described before.

- Delete the growth medium of a network in padmet format:

  ```
  aureme> aureme --run=test --cmd="del_medium NETWORK=network_name [NEW_NETWORK=new_
  ↪network_name]"
  ```

This function will remove all reactions consuming/producing a compound in ‘C-BOUNDARY’ compartment.

**Warning:** If you don’t precise any **NEW_NETWORK** name, the current network will be **overwritten**.

### 1.3.4 How to manage metabolic network compartment?

In a metabolic network a compound can occur in different compartment. Given a reaction transporting CA$^{2+}$ from ‘e’ (extra-cellular compartment) to ‘c’ (cytosol compartment), the compartments involved are ‘e’ and ‘c’. It is important to properly manage the compartments defined in a network to ensure a correct connection of the reactions. In some case metabolic networks can use different id to define a same compartment like ‘C_c’, ‘C’, ‘c’ for cytosol, merging those networks could leak to a loss of network connectivity. A reaction producing CA$^{2+}$ in ‘c’ and a reaction consuming CA$^{2+}$ in ‘C_c’ are actually not connected, hence the interest of the metabolic network compartment management commands of AuReMe.

- Get the complete list of compartment from a network in padmet format:

  ```
  aureme> aureme --run=test --cmd="get_compart NETWORK=network_name"
  ```

Return a list of compartment or an empty list.

- Change the id of a compartment from a network in padmet format:

  ```
  aureme> aureme --run=test --cmd="change_compart NETWORK=network_name OLD=old_id␣
  ↪NEW=new_id [NEW_NETWORK=new_network_name]"
  ```

This command will change the id of the compartment ‘**old_id**’ to ‘**new_id**’. This command is required if different ids are used to define a same compartment, example changing ‘C_c’ to ‘c’, or ‘C-c’ to ‘c’ . . .

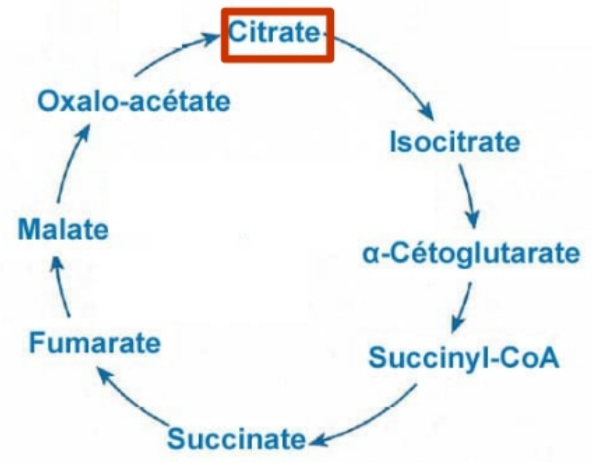
- Delete the id compartment from a network in padmet format:

```
aureme> aureme --run=test --cmd="del_compart NETWORK=network_name compart=compart_
↪id [NEW_NETWORK=new_network_name]"
```

This function will remove all reactions consuming/producing a compound in ‘**compart_id**’ compartment.

> **Warning:** If you don’t precise any **NEW_NETWORK** name, the current network will be **overwritten**.

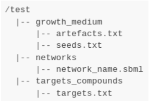
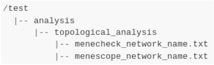
### 1.3.5 What are “artefacts”?

<table>
<tr>
<td>Meneco is a tool that fill the gaps topologically in a network, thanks to a reference database (see the <i>Method: Meneco</i> section). In fact, Meneco cannot product any other metabolite of an cycle without initiate it before.<br>Thereby, artefacts are metabolites allow Meneco to initiate cycles in a metabolic network.<br>For example in the picture aside, the Kreps cycle needs to be initiated with Meneco. A manner to initiate the Kreps cycle into Meneco is to put the “citrate” metabolite as one of the “artefacts”</td>
<td>


before gap-filling the network thanks to Meneco.</td>
</tr>
</table>

### 1.3.6 How to explore the topology of a metabolic network?

A manner of exploring and analyzing the topology of a metabolic network is to use the MeneTools (Metabolic Network Topology Tools). Two MeneTools: Menecheck and Menescope are included in AuReMe. You can run the one or the other individually.

| | |
|---|---|
| **Input files**  | **Result files**  |

```
/test
  |-- growth_medium
      |-- artefacts.txt
      |-- seeds.txt
  |-- networks
      |-- network_name.sbml
  |-- targets_compounds
      |-- targets.txt
```

```
/test
  |-- analysis
      |-- topological_analysis
          |-- menecheck_network_name.txt
          |-- menescope_network_name.txt
```

To obtain additional information about the file format of **artefacts.txt**, **seeds.txt**, and **targets.txt**, please refer to *Gap-filling input* and *What are "artefacts"?* sections.

- Menecheck gives the producibility status using graph-based criteria. To run Menecheck, use this command:

  ```
  aureme> aureme --run=test --cmd="menecheck NETWORK=network_name"
  ```

- Menescope provides the topologically reachable compounds from seeds (and artefacts) in a metabolic network. To run Menescope, use this command:

  ```
  aureme> aureme --run=test --cmd="menescope NETWORK=network_name"
  ```

### 1.3.7 How to manage the log files?

By default, the system registers all the executed commands as a log in the **bridge > test > log.txt** file. The whole output of these commands will also be stored in another file: the **bridge > test > full_log.txt** file.

If you DO NOT wish to store such logs, you can use the **quiet (-q)** argument in your command(s). For example:

```
aureme> aureme --run=test --cmd="some_commands" -q
```

It is possible to re-run a previous command by copying the corresponding command line in the **bridge > test > log.txt** file, and pasting it in the Docker container terminal.

To be able to reproduce the whole workflow applied in a previous study, please see the *How to reproduce studies?* section.

### 1.3.8 How to reproduce studies?

If you want to re-run the complete workflow of a pre-run study, built with AuReMe:

- First of all please create a new study (as described in the *Start a new study* section) by running the init command:

  ```
  aureme> aureme --init=my_run2
  ```

**Warning:** You can choose any run name you want, except pre-existing runs. Please, avoid other special characters than '_' and numbers).

It generates a new folder named **my_run2** in the **bridge** directory.

- Update your **config.txt** file, if it is needed.
- Now, copy all the input data from the previous study in this new folder (please, follow the folder architecture described in the *Data organization* section).
- Copy also the **log.txt** file in the **bridge > my_run2** directory, rename it (for example as run2.txt), and **change every occurrence of the previous run name by my_run2**.

- Execute the previously created file.

```
aureme> ./shared/my_run2/run2.txt
```

### 1.3.9 How to create a new 'à-la-carte' workflow?

If you want to add a new step in the workflow or add a new method, it is possible to customize AuReMe. For that it is necessary to update the Makefile in your run. Here is an example of how to do it.

- Add a new method:

First, install your tool by following the documentation associated. For the example we will add a new tool for orthology-based reconstruction 'new_tool' which use the same input as Pantograph (a metabolic network in sbml format, a gbk of the reference species and the gbk of the study species) and generate the same output (a metabolic network in sbml format).

Secondly we will update the Makefile by adding these lines:

Basically this command says that for each folder in orthology_based_reconstruction (variable declared in config.txt), if the expected output is not already created, run new_tool.

Finally, to select this method in your new workflow, change in the file config.txt the variable ORTHOLOGY_METHOD=pantograph by ORTHOLOGY_METHOD=new_tool

- Add a new step or function:

Just update the Makefile by adding a new step and use it with this command

### 1.3.10 How to choose another reference database?

It is possible to select a reference database among several. You can display the list of all available databases by using this command:

The reference database is needed to:

- be able to match all the identifiers of the entities of metabolic networks
- gap-fill the metabolic network in the gap-filling step

To select one, replace the corresponding path in the configuration file: ***config.txt***, in the ***DATA_BASE*** variable. Or you can comment the line if you don't want/can't use a database. The ***config.txt*** file is stored at the root of your ***bridge*** folder (see *Running a Docker container*).

### 1.3.11 What is checked in my input files?

Before running any command in AuReMe, it is highlight recommended to use the command 'check_input'. This command checks the validity of the input files and can also create required files. Concretely this command:

- Checks database: If database was specified in the config.txt file (see the *How to choose another reference database?* section). If so, checks if a sbml version exist and create it on the other hand.
- Checks studied organism data: Search if there is a genbank (gbk/gff) 'GBK_study.gbk' and proteome (faa) 'FAA_study.faa' in genomic_data folder. If there is only a genbank, create the proteome (command 'gbk_to_faa). If there is only the proteome or any of them, just continue the checking process. Note that the proteome is only required for the orthology-based reconstruction, method: Pantograph.

- |image8|Checks orthology-based reconstruction data: for each folder found in 'orthology_based_reconstruction' folder checks in each of them if there is proteome 'FAA_model.faa' and a metabolic network 'metabolic_model.sbml', if there is no proteome but a genbank file 'GBK_study.faa', create the proteome (command 'gbk_to_faa). Finally, the command compares the ids of genes/proteins between the proteome and the metabolic network.

If cutoff… important because… dict file to create a new proteome file …

- Checks annotation-based reconstruction data: for each folder found in annotation_based_reconstruction' folder checks in each of them if it's a PGBD from pathway then create (if not already done) a padmet file '**output_pathwaytools_**'folder_name'.padmet in networks/output_annotation_based_reconstruction folder.
- Checks gap-filling data: In order to gap-fill a metabolic network, Pantograph required as input, a file 'seeds.sbml' describing the seeds (the compounds available for the network), another describing the targets (the compounds that the network have to be able to reach), the metabolic network to fill and the database from where to draw the reactions all in sbml format. It's possible to start from txt files for seeds 'seeds.txt' and targets 'targets.txt', each file containing the ids of the compounds, one by line. The command will then convert them to sbml (command 'compounds_to_sbml').

Note that by default, AuReMe will integrate the artefacts 'default_artefacts_metacyc_20.0.txt' to the seeds to create a file 'seeds_artefacts.txt' and 'seeds_artefacts.sbml'. For more information about the artefacts see *What are "artefacts"?* section.

Example:

**[output] **

INSERT SCREEN FROM check_input log

### 1.3.12 What is the Makefile?

Makefile contient les cmd de AuReMe. exemple de cmd simple

### 1.3.13 What is the config.txt file?

The **config.txt** is found in the **bridge > test** directory. It contains all the AuReMe parameters: the name of the selected database, the name of the various choosen methods, and the default parameters of all programs that AuReMe needed.

If you want to use either another database or another tool already included in the AuReMe workspace, modify carefully the **config.txt** file.

> **Warning:** The parameters of the **config.txt** must not be changed unless you are sure of what you want do!

### 1.3.14 How to regenerate a new database version?

Voir les notes de Jeanne sur le problème de Sebastian

padmet/utils/connexion

### 1.3.15 How to map a metabolic network on another database?

Metabolic networks can be products of varied databases. If you want to merge efficiently information about metabolic networks coming from different databases, you will need to map the metabolic network(s) to a common database. To do so, a solution is provided be AuReMe.

Note: to use this method, the metabolic network to map needs to be in the SBML format and stored in the ***networks*** folder.

- First of all, you need to know the origin database of the data. To recognize the database used in an SBML file, use the ***which_db*** command:
  Example:

  **[output] **

- When you know the origin database of the data, you have to generate the mapping dictionary from this database to the new one:

  Example:

  **[output] **

  In this example, the system has found more than just one mapping for the *R_R00494_c* reaction and the *S_Starch_p* compound. It did not manage to choose between the propositions: the mapping will not be added to the output mapping. If you want to force the mapping, you have to modify the mapping file manually.

- Once you have created a mapping dictionary file, it will be automatically applied across the workflow to translate the data.

### 1.3.16 How to generate report on results?

Create reports on the **network_name.padmet** file network (in the **networks** directory).

```
aureme> aureme --run=test --cmd="report NETWORK=network_name"
```

Four files are created in the **analysis > reports > network_name** directory thanks to the report command.

- all_genes.csv (has the following format):

  ```
  id         Common name   linked reactions
  TL_15991   Unknown       2.3.1.180-RXN;RXN-9535
  TL_5857    Unknown       RXN-14271;RXN-2425
  TL_6475    Unknown       RXN-14229
  ```

  If a gene is linked with several reactions, reactions are separated from “;”.

- all_metabolites.csv (has the following format):

  ```
  dbRef_id       Common name         Produced (p), Consumed (c), Both (cp)
  NAD-P-OR-NOP   NAD(P)+             cp
  THIOCYSTEINE   thiocysteine        p
  CPD-18346      cis-vaccenoyl-CoA   c
  ```

- all_pathways.csv (has the following format):

  ```
  dbRef_id     Common name                             Number of reaction found  ␣
  ↪Total number of reaction    Ratio (Reaction found / Total)
  COA-PWY-1    coenzyme A biosynthesis II (mammalian)   1                         ␣
  ↪1                           1.00
  PWY-4984     urea cycle                               1                         ␣
  ↪5                           0.20
  PWY-7821     tunicamycin biosynthesis                 1                         ␣
  ↪9                           0.11
  ```

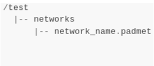
- all_reactions.csv (has the following format):

```
dbRef_id    Common name                                formula (with id)                ␣
↪                                formula (with common name)                      ␣
↪                                in pathways                     associated genes␣
↪            categories
NDPK        nucleoside-diphosphate kinase      1.0 ATP + 1.0 DADP => 1.0 ADP + 1.
↪0 DATP                           1.0 ATP + 1.0 dADP => 1.0 ADP + 1.0 dATP       ␣
↪                                                                 TL_16529;TL_
↪13128            ORTHOLOGY
RXN-15122   ORF                                1 THR => 1 PROTON + 1 CPD-15056 +␣
↪1 WATER                          1 L-threonine => 1 H+ + 1 (2Z)-2-aminobut-2-
↪enoate + 1 H2O                       PWY-5437;ILEUSYN-PWY;PWY-5826   TL_17207;TL_
↪12535;TL_8525    ANNOTATION;ANNOTATION;ORTHOLOGY
SGPL11      sphinganine 1-phosphate aldolase   1.0 CPD-649 => 1.0 PALMITALDEHYDE␣
↪+ 1.0 PHOSPHORYL-ETHANOLAMINE    1.0 sphinganine 1-phosphate => 1.0␣
↪palmitaldehyde + 1.0 O-phosphoethanolamine                                   TL_
↪105                       ORTHOLOGY
```

In this file, if there are several data in the same field, data are separated from “;”.

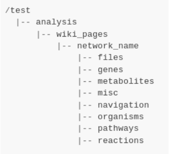
## 1.3.17 How to generate Wiki?

### Input files

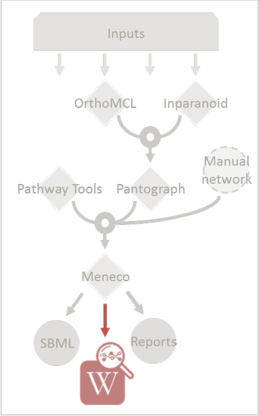


```
/test
  |-- networks
      |-- network_name.padmet
```

### Result files



```
/test
  |-- analysis
      |-- wiki_pages
          |-- network_name
              |-- files
              |-- genes
              |-- metabolites
              |-- misc
              |-- navigation
              |-- organisms
              |-- pathways
              |-- reactions
```



### Requirements

1. Utilize AuReMe, to create the wiki pages from a metabolic network.

An input file **network_name.padmet** inside the **brigde > test > networks** directory is needed. The wiki pages will be deployed in **brigde > test > analysis > wiki_pages > network_name**.

```
aureme> aureme --run=test --cmd="wiki_pages NETWORK=network_name"
```

> **Warning:** Run all the next commands from your machine and not from the AuReMe container.

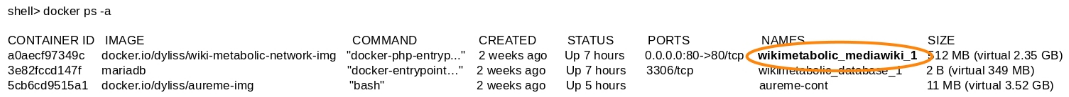
You can use wikis to analyze or visualize your metabolic networks, thanks to the MediaWiki technology.

2. Clone the wiki software within your computer:

```
shell> git clone https://github.com/AuReMe/wiki-metabolic-network.git
shell> cd wiki-metabolic-network/wiki-metabolic-network/
shell> make init
```

The **wiki-metabolic-network** is now installed on your computer. You can manage it in using the docker.com commands (see *Some tips about Docker*). wiki-metabolic-network is an image that allows to automatize the creation of wikis in a container.

- Get the name of the wiki container, it will be usefull to run the next command.



```
shell> docker ps -a

CONTAINER ID  IMAGE                                      COMMAND                 CREATED      STATUS      PORTS                 NAMES                        SIZE
a0aecf97349c  docker.io/dyliss/wiki-metabolic-network-img "docker-php-entryp..."  2 weeks ago  Up 7 hours  0.0.0.0:80->80/tcp    wikimetabolic_mediawiki_1    512 MB (virtual 2.35 GB)
3e82fccd147f  mariadb                                    "docker-entrypoint..."  2 weeks ago  Up 7 hours  3306/tcp              wikimetabolic_database_1     2 B (virtual 349 MB)
5cb6cd9515a1  docker.io/dyliss/aureme-img                "bash"                  2 weeks ago  Up 5 hours                        aureme-cont                  11 MB (virtual 3.52 GB)
```

> **Warning:** For a shake of genericity, in the following steps of this manual, we will employ the term of **wiki_cont** instead of **wikimetabolic_mediawiki_1** (the real one you have to write in your command lines).

- To enter the wiki container.

```
shell> docker exec -it wiki_cont bash
```

- To print the commands of the wiki container.

```
shell> docker exec -it wiki_cont wiki --help
```

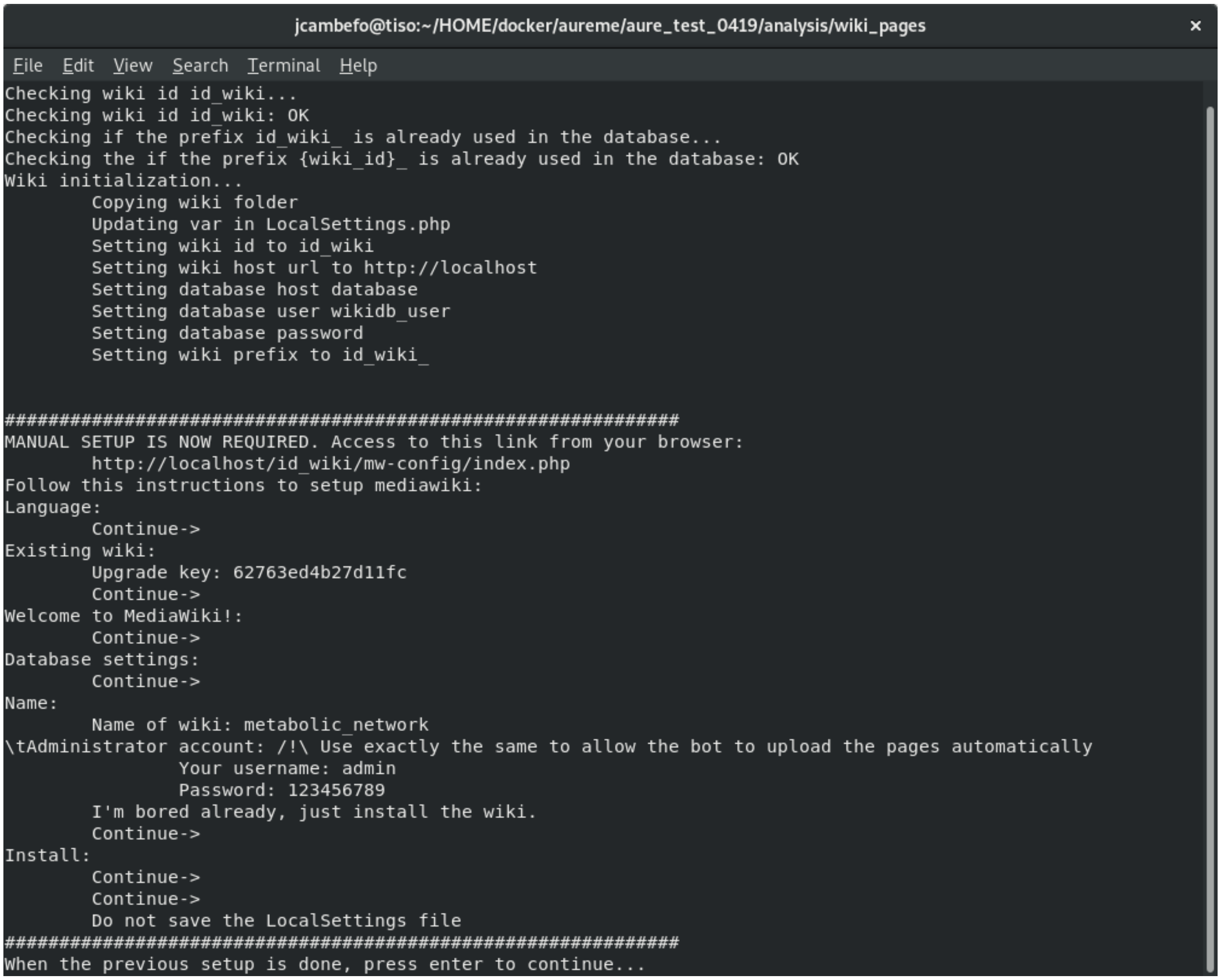
- Copy the data previously created thanks to AuReMe, in the wiki container.

```
shell> docker cp /test/analysis/wiki_pages/network_name wiki_cont:/home/
```

## Wiki creation

**Follow the instructions on your terminal.**

```
shell> docker exec -ti wiki_cont wiki --init=id_wiki
```

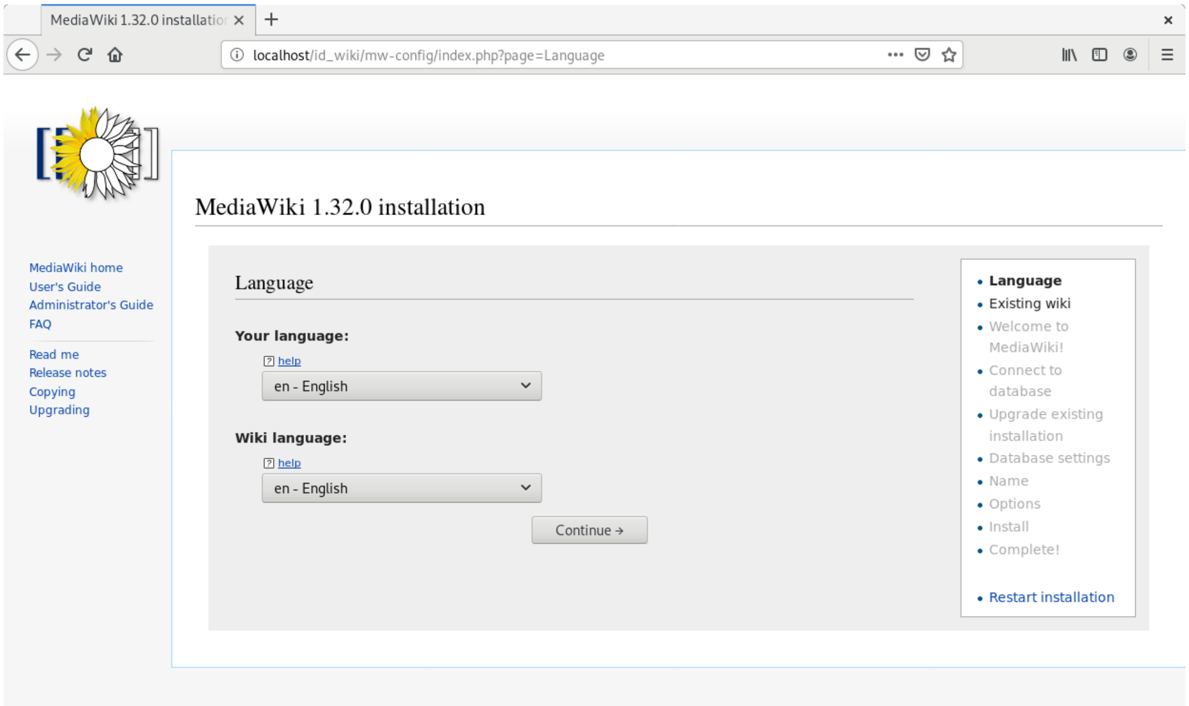


jcambefo@tiso:~/HOME/docker/aureme/aure_test_0419/analysis/wiki_pages
File Edit View Search Terminal Help

```
Checking wiki id id_wiki...
Checking wiki id id_wiki: OK
Checking if the prefix id_wiki_ is already used in the database...
Checking the if the prefix {wiki_id}_ is already used in the database: OK
Wiki initialization...
        Copying wiki folder
        Updating var in LocalSettings.php
        Setting wiki id to id_wiki
        Setting wiki host url to http://localhost
        Setting database host database
        Setting database user wikidb_user
        Setting database password
        Setting wiki prefix to id_wiki_


##############################################################
MANUAL SETUP IS NOW REQUIRED. Access to this link from your browser:
        http://localhost/id_wiki/mw-config/index.php
Follow this instructions to setup mediawiki:
Language:
        Continue->
Existing wiki:
        Upgrade key: 62763ed4b27d11fc
        Continue->
Welcome to MediaWiki!:
        Continue->
Database settings:
        Continue->
Name:
        Name of wiki: metabolic_network
\tAdministrator account: /!\ Use exactly the same to allow the bot to upload the pages automatically
                Your username: admin
                Password: 123456789
        I'm bored already, just install the wiki.
        Continue->
Install:
        Continue->
        Continue->
        Do not save the LocalSettings file
##############################################################
When the previous setup is done, press enter to continue...
```

1. Open your browser at the following address: **http://localhost/id_wiki/mw-config/index.php**, and press **“Continue”**.



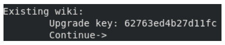
MediaWiki 1.32.0 installation

Language

**Your language:** en - English

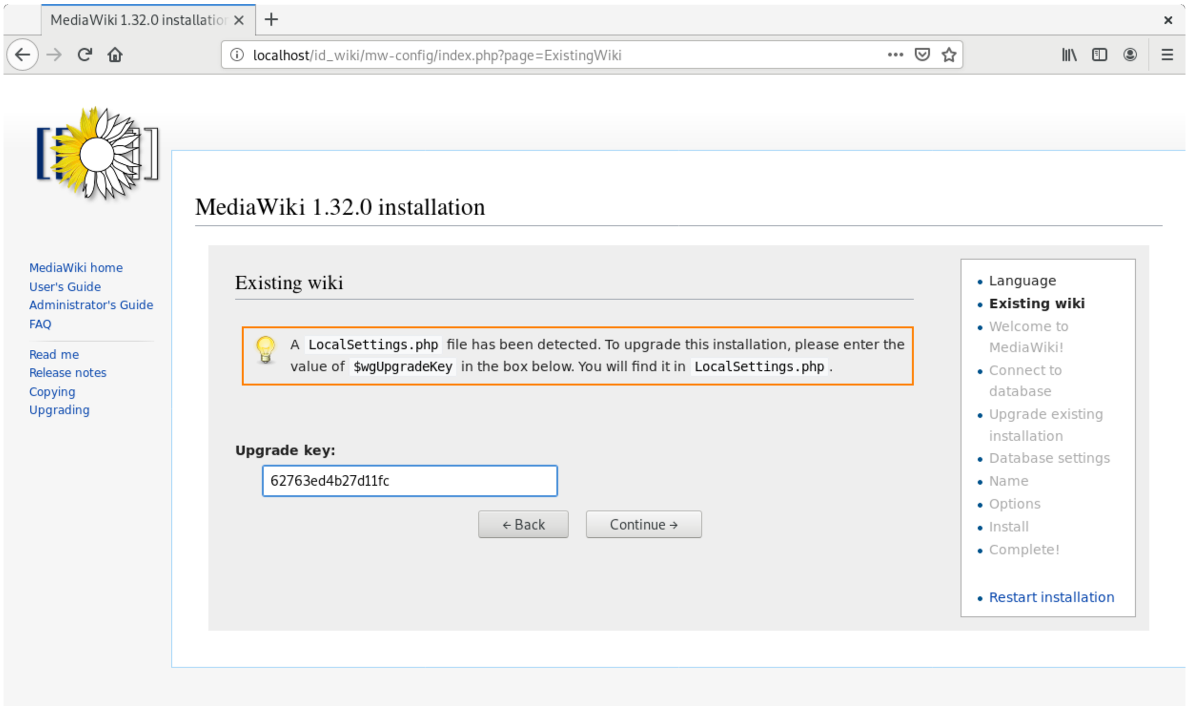
**Wiki language:** en - English

2. Get the **“Upgrade key”**. The **Upgrade key** is found on the your terminal. This is a small part extracted from the terminal to locate it better.



```
Existing wiki:
        Upgrade key: 62763ed4b27d11fc
        Continue->
```

3. Enter the **“Upgrade key”**, and press **“Continue”**.



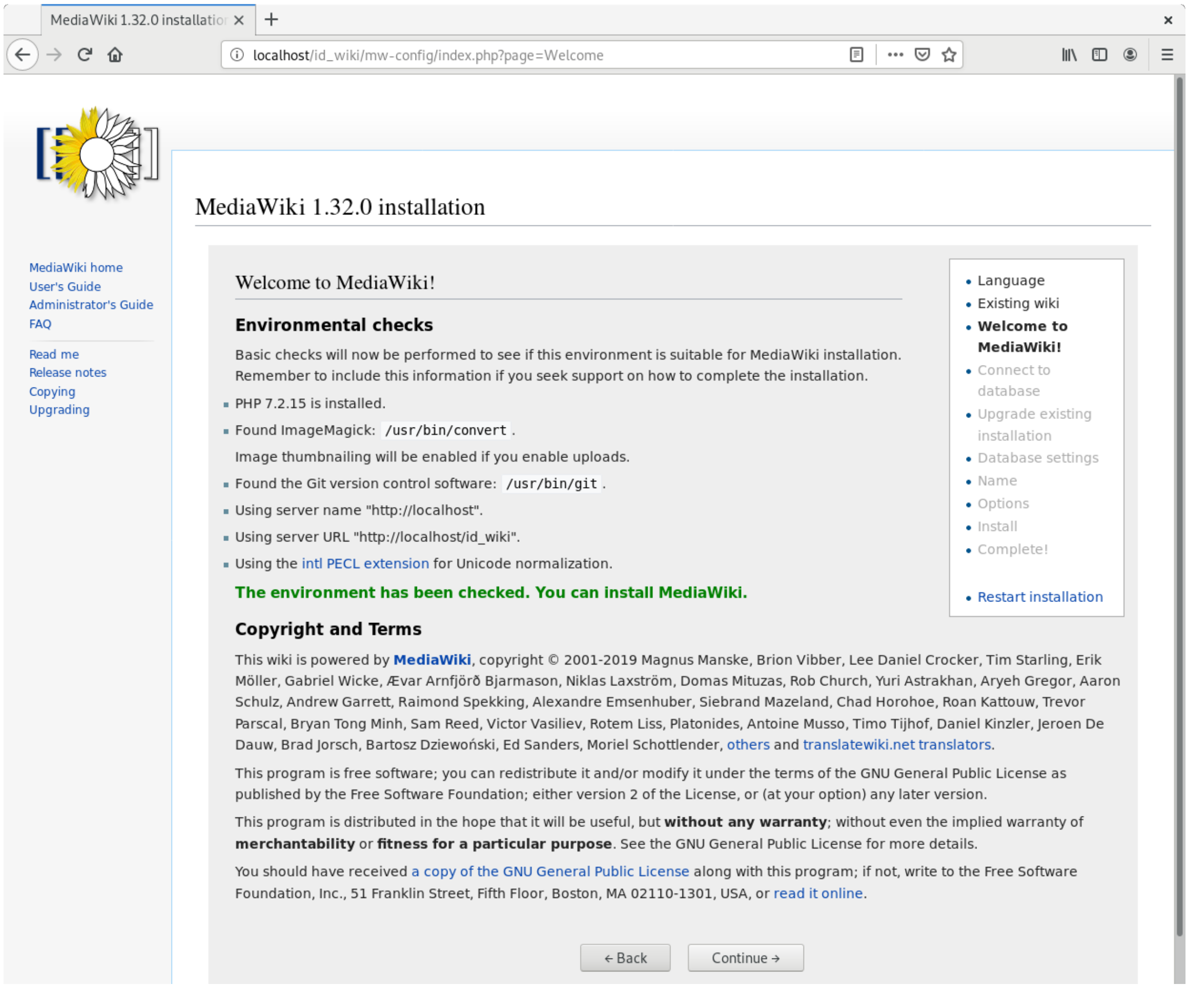
MediaWiki 1.32.0 installation

Existing wiki

A `LocalSettings.php` file has been detected. To upgrade this installation, please enter the value of `$wgUpgradeKey` in the box below. You will find it in `LocalSettings.php`.

**Upgrade key:** 62763ed4b27d11fc

4. In the page **“Welcome to MediaWiki”** configuration, just press **“Continue”**.



MediaWiki 1.32.0 installation

Welcome to MediaWiki!

**Environmental checks**

Basic checks will now be performed to see if this environment is suitable for MediaWiki installation. Remember to include this information if you seek support on how to complete the installation.

- PHP 7.2.15 is installed.
- Found ImageMagick: `/usr/bin/convert`.
  Image thumbnailing will be enabled if you enable uploads.
- Found the Git version control software: `/usr/bin/git`.
- Using server name "http://localhost".
- Using server URL "http://localhost/id_wiki".
- Using the intl PECL extension for Unicode normalization.

**The environment has been checked. You can install MediaWiki.**

**Copyright and Terms**

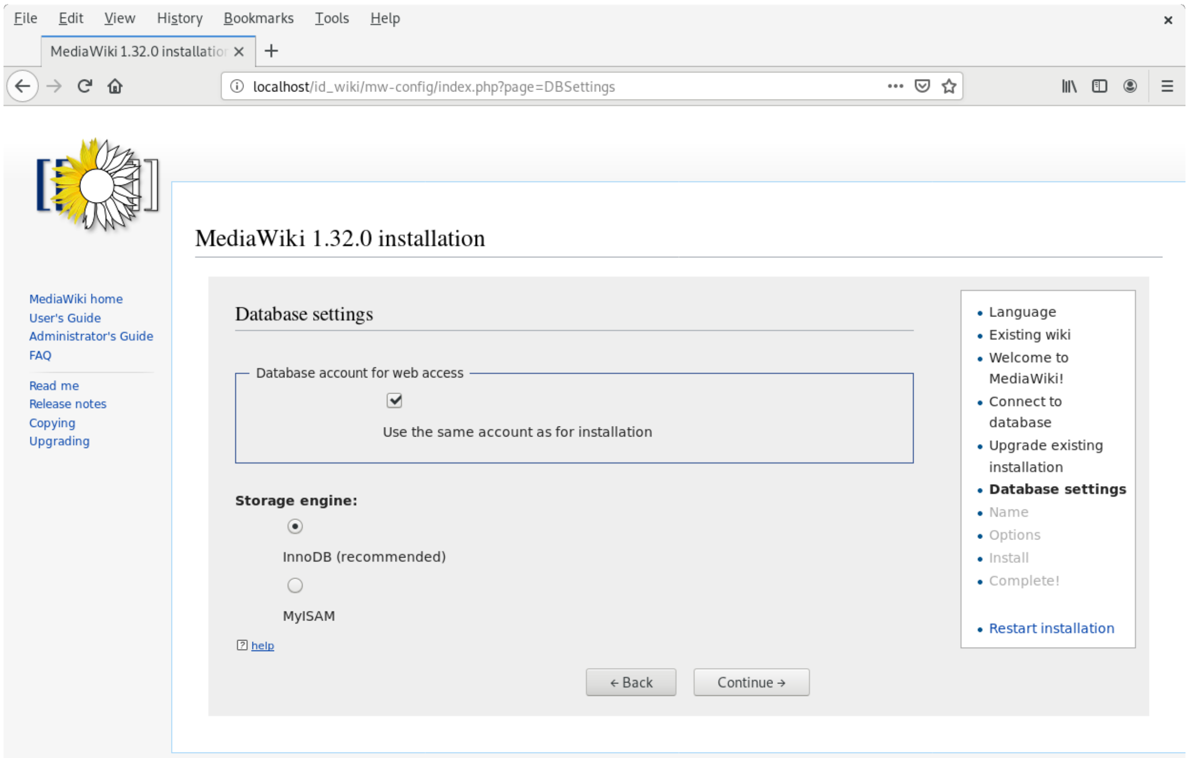
This wiki is powered by **MediaWiki**, copyright © 2001-2019 Magnus Manske, Brion Vibber, Lee Daniel Crocker, Tim Starling, Erik Möller, Gabriel Wicke, Ævar Arnfjörð Bjarmason, Niklas Laxström, Domas Mituzas, Rob Church, Yuri Astrakhan, Aryeh Gregor, Aaron Schulz, Andrew Garrett, Raimond Spekking, Alexandre Emsenhuber, Siebrand Mazeland, Chad Horohoe, Roan Kattouw, Trevor Parscal, Bryan Tong Minh, Sam Reed, Victor Vasiliev, Rotem Liss, Platonides, Antoine Musso, Timo Tijhof, Daniel Kinzler, Jeroen De Dauw, Brad Jorsch, Bartosz Dziewoński, Ed Sanders, Moriel Schottlender, others and translatewiki.net translators.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but **without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA, or read it online.

5. In the page **"Database settings"** configuration, just press **"Continue"**.



localhost/id_wiki/mw-config/index.php?page=DBSettings

MediaWiki 1.32.0 installation

Database settings

Database account for web access

☑ Use the same account as for installation

**Storage engine:**

● InnoDB (recommended)

○ MyISAM

6. In the page **“Name”** configuration, you have several fields to fill:

   a. Name of wiki: **wiki_name**

   b. Your username: **admin**

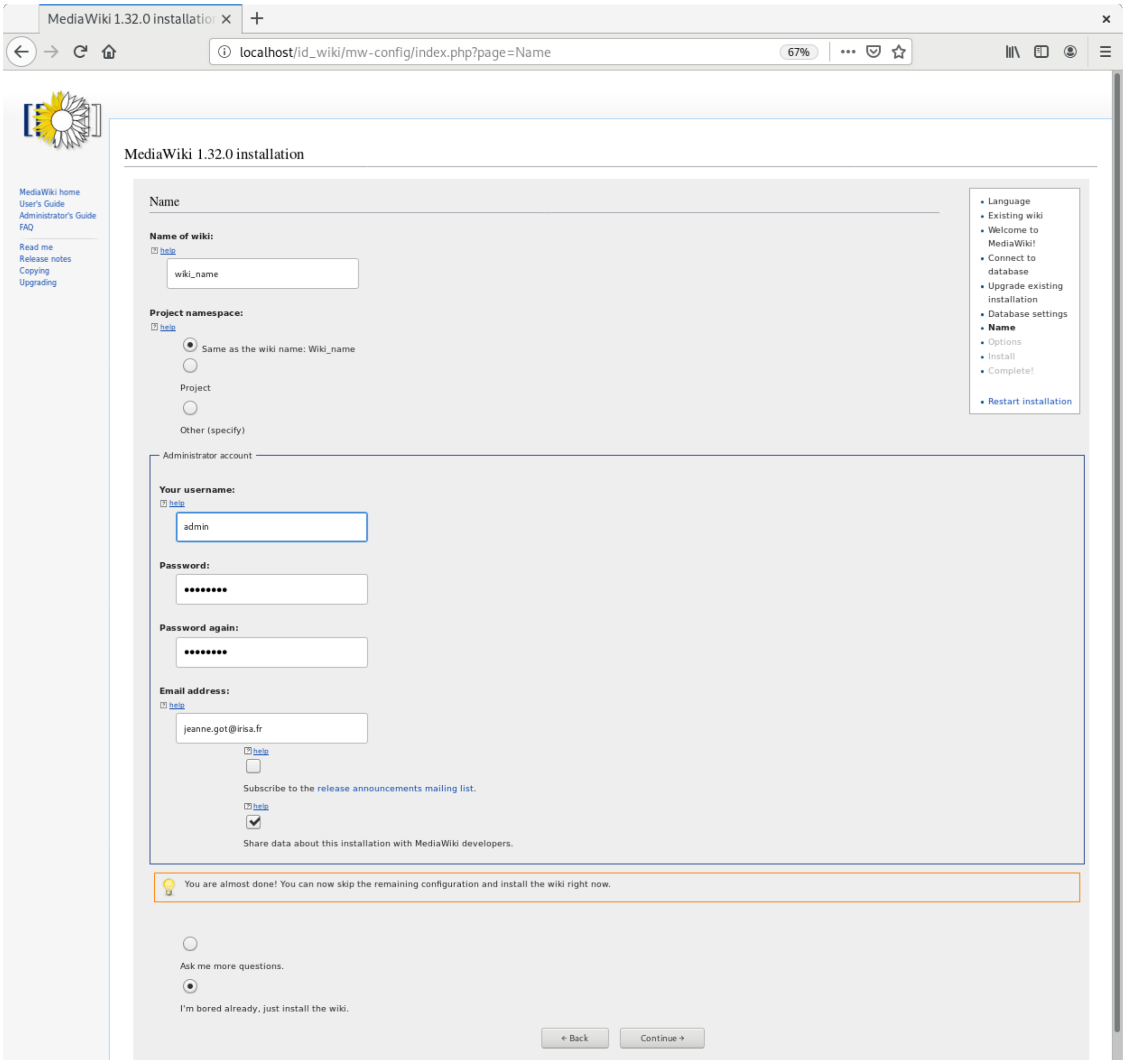
   c. Password: Enter a password (it is at least 8 characters).

   d. Password again: Enter the same password.

   e. Email address: jeanne.got[at]irisa.fr (for example)

   f. Please select the phrase: **“I’m bored already, just install the wiki”**.

   g. Press **“Continue”**.



MediaWiki 1.32.0 installation

Name

Name of wiki: wiki_name

Project namespace:

● Same as the wiki name: Wiki_name

○ Project

○ Other (specify)

Administrator account

Your username: admin

Password: ••••••••

Password again: ••••••••

Email address: jeanne.got@irisa.fr

□ Subscribe to the release announcements mailing list.

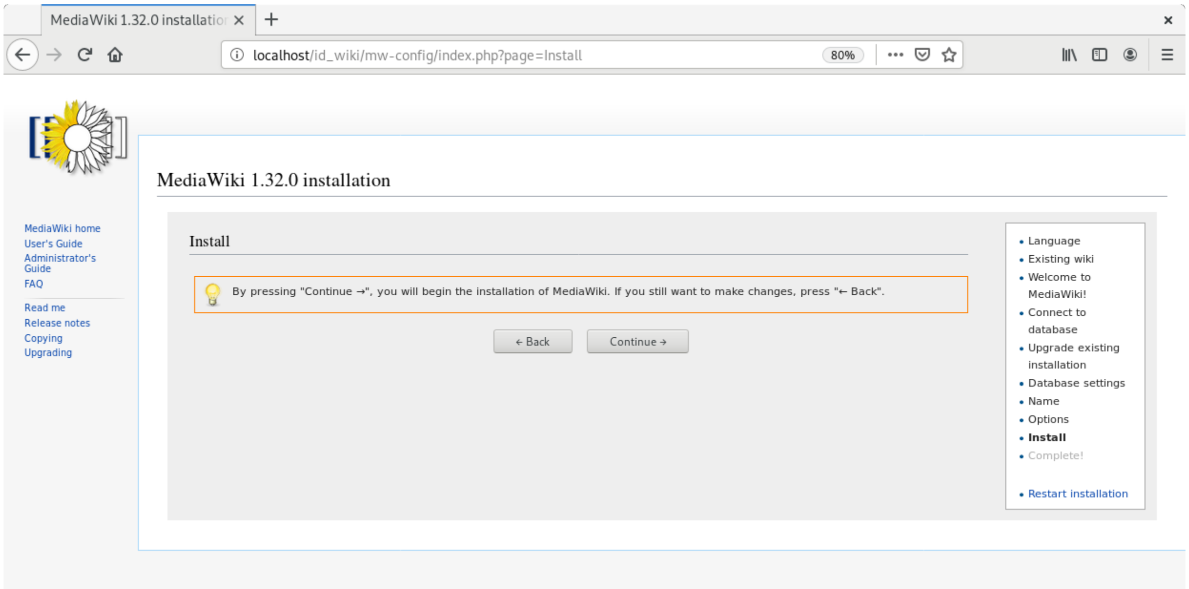
☑ Share data about this installation with MediaWiki developers.

You are almost done! You can now skip the remaining configuration and install the wiki right now.

○ Ask me more questions.

● I'm bored already, just install the wiki.

7. In the first page of **"Install"**, just press **"Continue"**.

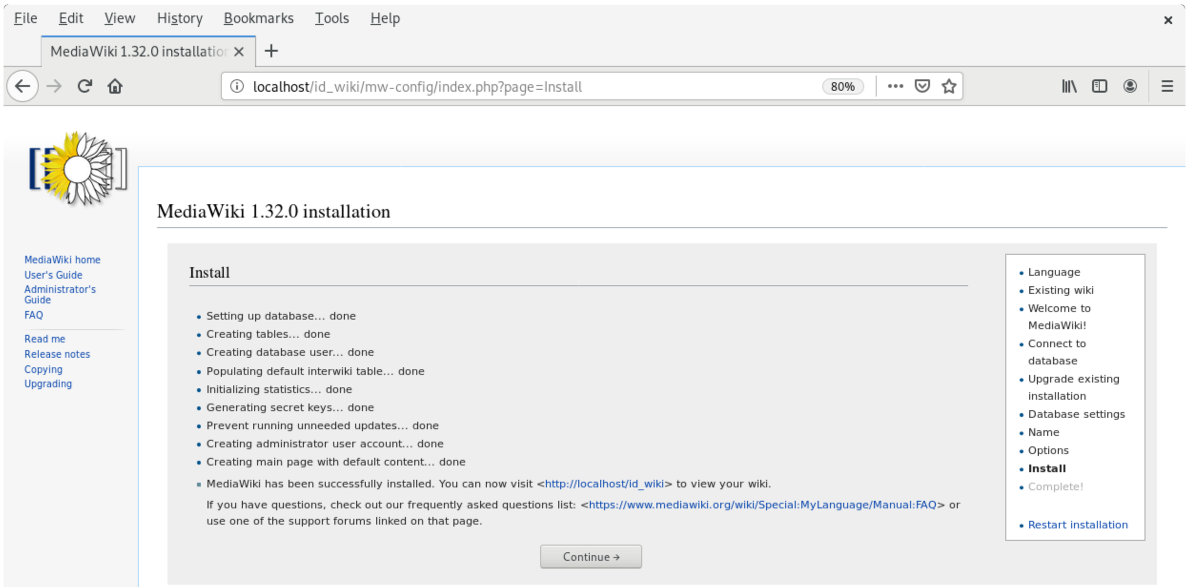


MediaWiki 1.32.0 installation

Install

By pressing "Continue →", you will begin the installation of MediaWiki. If you still want to make changes, press "← Back".

8. In the second page of **"Install"**, just press **"Continue**.



MediaWiki 1.32.0 installation

Install

- Setting up database... done
- Creating tables... done
- Creating database user... done
- Populating default interwiki table... done
- Initializing statistics... done
- Generating secret keys... done
- Prevent running unneeded updates... done
- Creating administrator user account... done
- Creating main page with default content... done
- MediaWiki has been successfully installed. You can now visit <http://localhost/id_wiki> to view your wiki.
  If you have questions, check out our frequently asked questions list: <https://www.mediawiki.org/wiki/Special:MyLanguage/Manual:FAQ> or use one of the support forums linked on that page.

9. Do not download the LocalSettings.php file.

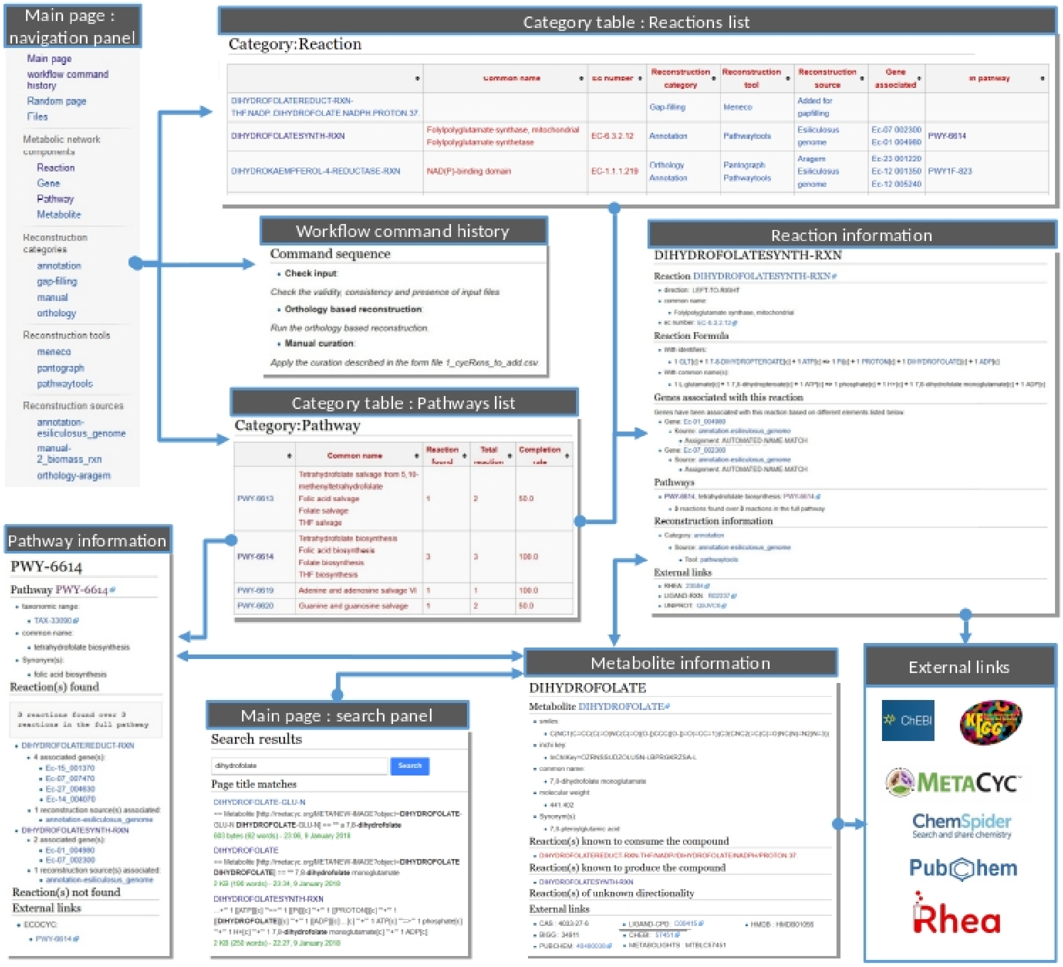
10. **Go back to your terminal, and press "Enter".** The wiki is now online and reachable at this link: **http://localhost/id_wiki/index.php/Main_Page**.

- To send the "wiki pages" (that you previously copied in the wiki_cont container) on the wiki.

```
shell> docker exec -ti wiki_cont wiki_load --action=load --url=http://localhost/
↪id_wiki/api.php --user=admin --password=my_password --wikipage=/home/network_
↪name --bots=2 -v
```

  Here **"bots"** is the number of CPUs are allocated to make this task.

- Now wiki pages are accessible on **http://localhost/id_wiki/index.php**. The following picture shows some functionalities of the wiki.



## Public and private access

**Note:** By default, a **"public access"** wiki is created. A wiki with a public access means, a wiki which everyone is allowed to access and to edit it on condition that she/he has an account on this wiki.

- To deploy a wiki with a **"private access"**.

```
shell> docker exec -ti wiki_cont wiki --init=id_wiki --access=private
```

  Then, see the *Wiki creation* section. A wiki with a **"private access"** is preventing access and editing for non-user. It also prevent account creation. It is useful to manage confidential data.

- To modify the access of a wiki already created.

```
shell> docker exec -ti wiki_cont wiki --id=id_wiki --access=private
shell> docker exec -ti wiki_cont wiki --id=id_wiki --access=public
```

#### Other wiki commands

- To list all deployed wiki use.

```
shell> docker exec -ti wiki_cont wiki --all
All deployed wiki:
        C_elegans
        E_siliculosus
        id_wiki
        S_cerevisiae
```

- To remove a wiki use.

```
shell> docker exec -ti wiki_cont wiki --id=id_wiki --remove
Removing wiki id_wiki
Removing wiki folder
97 tables to drop
```

  It removes the "id_wiki" from wiki_folders and removes tables from database which start with prefix id_wiki.

- To reset a wiki use.

```
shell> docker exec -ti wiki_cont wiki --id=id_wiki --clean
id_wiki_page table to empty
```

  It only remove all the pages of the specified wiki. It keeps tables and folder associated with this wiki.

### 1.3.18 How to connect to Pathway-tools?

- Create PGDB from output of AuReMe

### 1.3.19 How to set an objective reaction?

To add a biomass reaction to a network, see the *Create new reaction(s) to add in a network* section. Once the biomass is included in the network, you have to set the biomass as objective function.

Apply this command to the **network_name.padmet**

```
aureme> aureme --run=test --cmd="set_fba ID=reaction_name NETWORK=network_name"
```

It creates the **network_name.sbml** file with reaction_name as the objective function. To continue the analyzis of the network_name, see the *How to process Flux Balance Analysis?* section.

### 1.3.20 How to process Flux Balance Analysis?

AuReMe evaluate the flux balance analyzis of the biological network, thanks to the cobrapy Python package. Before calculating the flux balance analysis of a network:

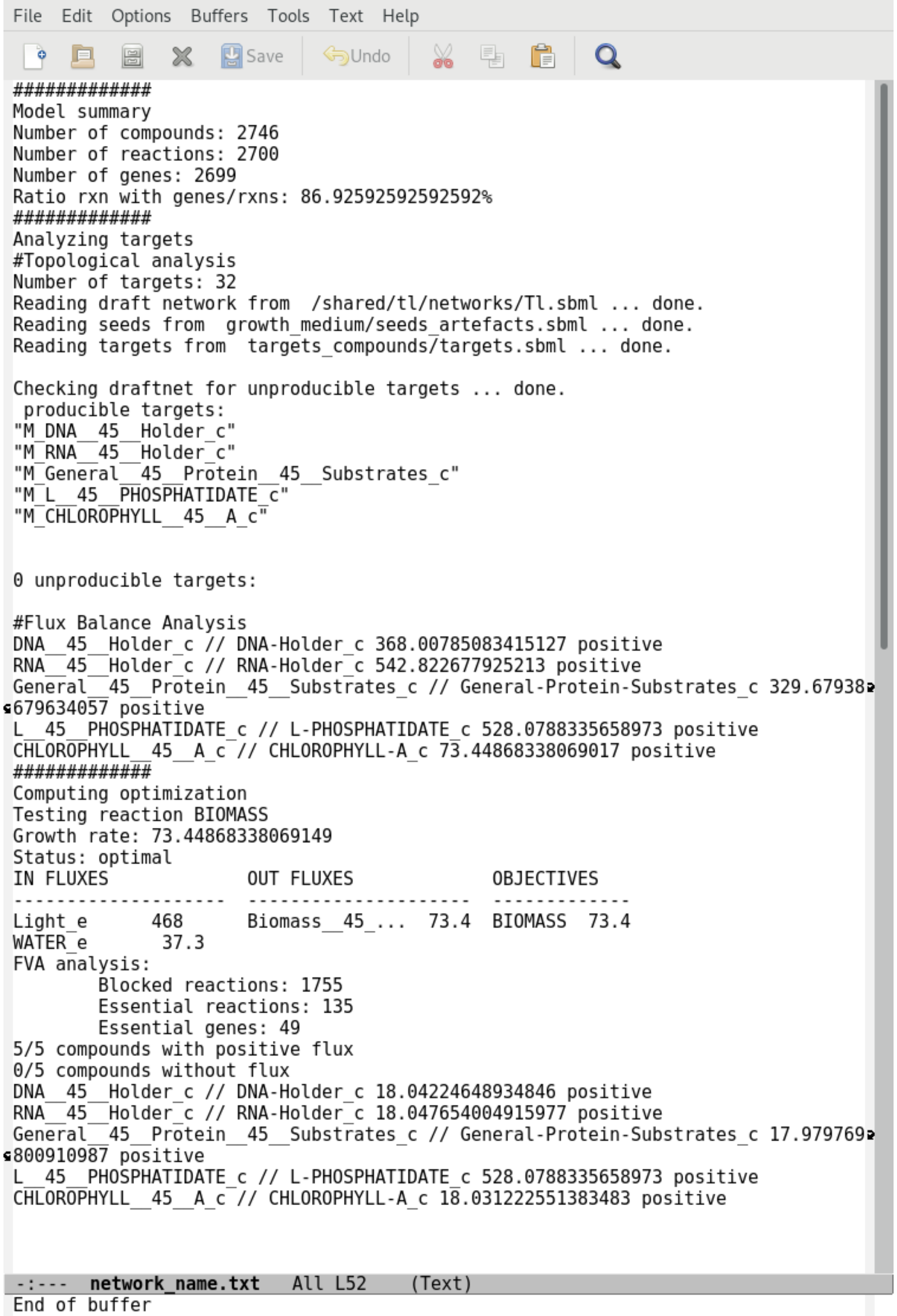
a. you may have to add the biomass to a network in reporting to the *Create new reaction(s) to add in a network* section,

b. you have to set the biomass as an objective reaction, please refer to the *How to set an objective reaction?* section.

To compute the flux balance analyzis of the **network_name.sbml** file:

```
aureme> aureme --run=test --cmd="summary NETWORK=network_name"
```

Two files: **network_name.txt** and **network_name_log.txt** are generated in the **analysis > flux_analysis** directory. The first file (**network_name.txt**) summarizes te network, then it get the list of productible and unproductible targets. For each productible target, the flux balance analysis is given. The growth rate of the network is also provided. Here is an example of a **network_name.txt** format:



```
#############
Model summary
Number of compounds: 2746
Number of reactions: 2700
Number of genes: 2699
Ratio rxn with genes/rxns: 86.92592592592592%
#############
Analyzing targets
#Topological analysis
Number of targets: 32
Reading draft network from  /shared/tl/networks/Tl.sbml ... done.
Reading seeds from  growth_medium/seeds_artefacts.sbml ... done.
Reading targets from  targets_compounds/targets.sbml ... done.

Checking draftnet for unproducible targets ... done.
 producible targets:
"M_DNA__45__Holder_c"
"M_RNA__45__Holder_c"
"M_General__45__Protein__45__Substrates_c"
"M_L__45__PHOSPHATIDATE_c"
"M_CHLOROPHYLL__45__A_c"


0 unproducible targets:

#Flux Balance Analysis
DNA__45__Holder_c // DNA-Holder_c 368.00785083415127 positive
RNA__45__Holder_c // RNA-Holder_c 542.822677925213 positive
General__45__Protein__45__Substrates_c // General-Protein-Substrates_c 329.67938
679634057 positive
L__45__PHOSPHATIDATE_c // L-PHOSPHATIDATE_c 528.0788335658973 positive
CHLOROPHYLL__45__A_c // CHLOROPHYLL-A_c 73.44868338069017 positive
#############
Computing optimization
Testing reaction BIOMASS
Growth rate: 73.44868338069149
Status: optimal
IN FLUXES             OUT FLUXES             OBJECTIVES
--------------------  ---------------------  -------------
Light_e      468      Biomass__45_...  73.4  BIOMASS  73.4
WATER_e       37.3
FVA analysis:
        Blocked reactions: 1755
        Essential reactions: 135
        Essential genes: 49
5/5 compounds with positive flux
0/5 compounds without flux
DNA__45__Holder_c // DNA-Holder_c 18.04224648934846 positive
RNA__45__Holder_c // RNA-Holder_c 18.047654004915977 positive
General__45__Protein__45__Substrates_c // General-Protein-Substrates_c 17.979769
800910987 positive
L__45__PHOSPHATIDATE_c // L-PHOSPHATIDATE_c 528.0788335658973 positive
CHLOROPHYLL__45__A_c // CHLOROPHYLL-A_c 18.031222551383483 positive



-:---  network_name.txt   All L52    (Text)
End of buffer
```

The second file (**network_name_log.txt**) supplies all the warnings produced computing the flux balance analyzis.

## 1.4 Indices and tables

- genindex
- modindex
- search