Minotour Documentation

Release 1.0

Matt Loose, Roberto Santos, Rory Munro

March 26, 2019

CONTENTS:

CHAPTER

ONE

USING DOCKER

Installation

First, if not already installed, install docker community edition, following the instructions found here.

Check the installation was correct, either in the terminal, or PowerShell for windows:

```
docker --version
```

Clone the repository into an appropriate location, or download and unzip the file. In the terminal or PowerShell change directory into the new minotourapp directory.:

```
cd /path/to/minotourapp/
```

Build and run the docker containers. There is a helpful script that will do this for you. In UNIX environments (linux or mac) run:

```
sudo ./docker-config.sh
```

On windows, run:

```
./docker-config.bat
```

Once this script has finished running, minotour should be up and accessible from localhost:10000.

Create an administration account

First to create a user with administrator rights:

```
docker ps -a
```

This will show you all the containers you have installed. Copy the container ID for the web-minotour container. Run:

```
docker exec -it <Container ID> python3 manage.py createsuperuser
```

Follow the prompts, filling in Username, Email and Password.

Add references for mapping/metagenomics

Now, in order to add references to the docker container for mapping and metagenomics it is necessary to add the reference fasta files into the celery-worker container.

To do this, run the following commands:

```
docker ps -a
```

Copy the container ID for the celery-worker-minotour-instance container. Run:

```
docker cp /path/to/folder/containing/reference/files/. <Container ID>:/var/lib/minotour/data
```

Now to add references to the worker application itself:

```
docker exec <Container ID> python3 manage.py add_references /var/lib/minotour/data
```

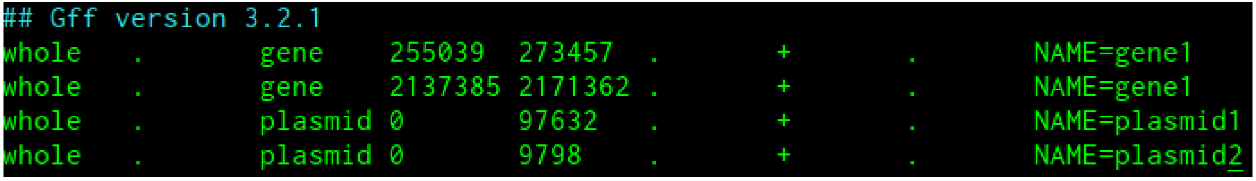
Add the centrifuge index

To add the centrifuge index, you will need to have the files locally to copy across into the celery-worker-minotour-instance container.:

```
docker cp /path/to/folder/containing/index/files/. <Container ID>:/var/lib/minotour/data
```

Add the validation regions

Validation regions must be created as GFF3 files, following the universal format found here. An example file looks like



```
## Gff version 3.2.1
whole   .       gene    255039  273457  .       +       .       NAME=gene1
whole   .       gene    2137385 2171362 .       +       .       NAME=gene1
whole   .       plasmid 0       97632   .       +       .       NAME=plasmid1
whole   .       plasmid 0       9798    .       +       .       NAME=plasmid2
```

Finally, to add a set of validation regions:

```
docker ps -a
```

Copy the container ID for the celery-worker-minotour-instance container. Then run:

```
docker cp /path/to/folder/containing/gff/files/. <Container ID>:/var/lib/minotour/data
```

And to add them to the application:

```
docker exec <Container ID> python3 manage.py add_validation_sets -S <desired_set_name> -k <api_key> /
```

The api key can be found on the profile section of a logged in account on the minotour page, under the drop down of the username in the top nav bar.

CHAPTER

TWO

MINOTOUR CLIENT

The client is dependent on python3.5 and above. We recommend creating a virtual environment to contain the environment and avoid polluting your global environment.:

```
python3 -m venv /path/to/envs/minfq
```

Activate the environment:

```
source /path/to/envs/minfq/bin/activate
```

Upgrade pip to its latest version:

```
pip install --upgrade pip
```

The client is available on PyPi:

```
pip install minFQ
```

A development version is also available. Clone the client repository:

```
git clone https://github.com/LooseLab/minotourcli.git
```

Or download and unzip the code found in the github repository.

To install the development version into the python virtual environment:

```
pip install -e .
```

Check the client is installed into the virtual environment (make sure the environment is activated, using the above source command):

```
minFQ -h
```

You should see a helpful help page.

To upload data to the minotour application, you will need the API key of the registered user you wish the data to be kept under. This can be found in the profile section of this user.

To access the profile page, login using your username and password, click the username in the top nav bar, click the profile option. Copy the API key to clipboard.

An example minFQ command would be:

```
minFQ -w /path/to/directory/containing_fastq/ -n <flowcell_name> -k <Api_key> -hn <Server address fo
```

For docker the port is 10000. The host server address for minotour is localhost, or 127.0.0.1. The minKNOW ip address is usually 127.0.0.1. The flowcell name can be what you wish, and the watch directory is where the basecalled Fastq will be appearing.

minFQ scans recursively, so setting -w a top level directory for the flowcell data will find all the runs and Fastqs inside that directory.

For the development environment, this will be similar, but the port can vary.

For the minotour web server hosted by Nottingham, the host server address for minotour will be minotour.nottingham.ac.uk.

Config file

minFQ can be preconfigured with a lot of the options that stay the same using a config file, that must be present in the Current Working Directory that minFQ is being called from.

An example config file could be called minfq-posix.config, and any option can be configured using the – name of that argument. Config file syntax allows: key=value, flag=true, stuff=[a,b,c].

The content of a config file may look like:

```
key=b410a0c9729d92ac989509c695cc3ee66a749ec6
port=8000
hostname=127.0.0.1
ip-address=127.0.0.1
```

CHAPTER

THREE

DEVELOPMENT ENVIRONMENT

To setup a local development environment, first clone or download and unzip the code from github found here.

Checkout the development branch:

```
git checkout develop
```

To run the python package mysqlclient, it is often necessary to have the following two dependencies installed, libmysqlclinet-dev and python3-dev. These can be installed with the following command:

```
sudo apt-get install libmysqlclient-dev
sudo apt-get install python3-dev
```

Create a virtual environment for the project dependencies and install them:

```
cd /path/to/minotourapp/code
python3 -m venv minotourenv
source minotourenv/bin/activate
pip install -r requirements.txt
```

Minotour contains three main modules:

1. The client, currently minFQ
2. web, the interface that most end users have access and provides access to active and archived runs
3. the rest api, that is a gateway connecting client and web modules to data and core Minotour functionalities.

Minotour also makes use of a MySQL database, Celery (responsible for running server tasks), Redis (a database in memory similar to memcached, required by Celery as a message broker), Flower (optional - but a useful Celery task monitor tool).

- [Redis] - Minotour uses **Redis** as a cache system for the web module and also for Celery. Follow the instructions and make sure that the **redis-server** executable is available in PATH environment variable.
- [MySQL Community edition] - Minotour requires a MySQL server instance. It can run locally or on another server. Installing and configuring MySQL is not in the scope of this guide, but here is the official documentation and another good tutorial can be found here. Once the server is up and running you can either choose to use the root user created during mySql initialisation in the below environmental variables, or create a user as follows in the mysql shell, logged in as the root user or an admin user:

```
    CREATE USER minotour;
    CREATE TABLE minotourdb;
    GRANT ALL PRIVILEGES ON minotourdb TO 'minotour'@'localhost' IDENTIFIED BY '<password>'
```

- [Minimap2] - Minotour uses **Minimap2** to run fast alignment, and to do metagenomics target validation.
- [Python 3] - Minotour uses **Python >=3.5**, so make sure it is available on your system.

- You will need to have the virtual environment activated to run Minotour, Celery and Flower, as well as to install the new dependencies.
- Create Minotour data directory - currently, Minotour uses this folder to keep track of the genome references available:

```
mkdir -p ~/data/minotour;
```

Metagenomics analyses

minoTour uses Centrifuge to run metagenomics analyses. Before creating new metagenomics tasks, there are a few requirements that need to be completed.

- Centrifuge index - Choose one of the centrifuge indexes (ftp://ftp.ccb.jhu.edu/pub/infphilo/centrifuge/data) and save in the local disk. We recommend the compressed index p_compressed
- Centrifuge application - Download and compile the most recent version.
- Set the environment variables MT_CENTRIFUGE (the path to the executable) and MT_CENTRIFUGE_INDEX (the path to indexes without the suffix) as below:

```
export MT_CENTRIFUGE="/home/user/centrifuge-1.0.4-beta/centrifuge"

export MT_CENTRIFUGE_INDEX="/home/user/centrifuge_indexes/p_compressed"
```

The MT_CENTRIFUGE_INDEX above points to the following indexes:

p_compressed.1.cf

p_compressed.2.cf

p_compressed.3.cf

p_compressed.4.cf

- minoTour uses the ete3 package, that needs access to the internet to download the NCBI species database.

To force the download (this step just need to be executed once), type the following instructions on the python interpreter:

```
import ete3

from ete3 import NCBITaxa

n = NCBITaxa()
```

Environmental config and running

- Setup environment variables - many Minotour config parameters are stored in the environment, so that we do need to hardcode database users and password, or any other information that is environment dependent. You can create a bash script file, for example envs.sh, place it in the main application directory (The directory that contains manage.py) and copy the following into it, or include them in the ~/.bash_profile or ~/.bash_rc (please check if this is the correct file in your environment):

```
#!/bin/bash
export MT_DB_ENGINE='django.db.backends.mysql'
export MT_DB_HOST='localhost'
export MT_DB_NAME='minotourdb'
export MT_DB_PASS='<minotourdb password>'
export MT_DB_PORT='3306'
export MT_DB_USER='minotour'
export MT_DJANGO_DEBUG='True'
export MT_MAILGUN_ACCESS_KEY=''
export MT_MAILGUN_SERVER_NAME=''
export MT_MINIMAP2='/usr/bin/minimap2'
export MT_REFERENCE_LOCATION='/home/rory/data/minotour'
export MT_SECRET_KEY=''
export MT_TWITCONSUMER_KEY=''
export MT_TWITCONSUMER_SECRET=''
export MT_CENTRIFUGE="</path/to/centrifuge/executable>"
export MT_CENTRIFUGE_INDEX="/path/to/centrifuge/indexes"
export MT_TWITTOKEN=''
export MT_TWITTOKEN_SECRET=''
export MT_LOG_FOLDER='/path/to/where/you/want/logs'
export MT_CELERY_BROKER_URL='redis://localhost:6379/0'
export MT_CELERY_RESULT_BACKEND='redis://localhost:6379/0'
```

- Now it is time to start the processes, we suggest opening a new terminal for each command. If you chose to create the environmental variable bash file, add the following to the beginning of the celery, flower and Minotour commands to set the environmental variables:

```
. envs.sh &&
```

- MySQL - make sure it is running **AND the database was created** (check the docs mentioned above).
- Redis:

```
redis-server &
```

- Create tables and administrator account:

```
cd /path/to/minotour/code;
source ~/minotourenv/bin/activate;
python3 manage.py makemigrations;
python3 manage.py migrate;
python3 manage.py loaddata fixtures/auxiliary_data.json
python3 manage.py createsuperuser
```

- Start Celery:

```
cd /path/to/minotour/code/ && source minotourenv/bin/activate && celery -A minotourapp worker -l
```

- Start Flower:

```
cd /path/to/minotour/code/ && source minotourenv/bin/activate && flower -A minotourapp --port=55
```

- Start Minotour:

```
cd /path/to/minotour/code
source minotourenv/bin/activate
python manage.py runserver 8100
```

- Time to test - if everything worked well, you should be able to access the web interface on http://localhost:8100.
- To add references and validation sets:

```
python3 manage.py add_references /path/to/reference/files/directory
```

- Finally, to add a set of validation regions, in the format of gff3:

```
python3 manage.py add_validation_sets -S <desired_set_name> -k <api_key> /var/lib/minotour/data
```

The api key can be found on the profile section of a logged in account on the minotour page, under the drop down of the username in the top nav bar.

CHAPTER

FOUR

QUICKSTART USE CHEATSHEET

Docker

If the docker containers have been setup following the instructions here - Using docker

Then to start the containers after restarting a computer, on unix simply run:: sudo ./docker-start.sh

And on windows simply run:: ./docker-start.bat

To stop containers - run the respective docker-stop file.

Development

If the minotourapp has been installed correctly, simply run:

```
python3 manage.py runserver 8100
```

And in a seperate terminal:

```
cd /path/to/minotour/code/ && source minotourenv/bin/activate && celery -A minotourapp worker -l info
```

CHAPTER

FIVE

FAQ

1. How much space will I require on my system?

   Whilst the application itself is a few hundred megabytes, it stores all the results in the backend database. Alongside this, the standard centrifuge index is 8Gb itself. So we recommend at least 300Gb of space if you are using Minotour for full analysis, and not just monitoring.

   If you plan to just use minoTour for monitoring, the size requirements will be dramatically reduced.

2. Is a docker hosted system accessible from other computers on the same network?

   No. Not without port forwarding setup on incoming traffic. How to do this is outside the scope of this documentation.

3.