

# easyMF User Manual

---

(version 1.0)

easyMF is a user-friendly web platform that aims to facilitate biological discovery from large-scale transcriptome data through matrix factorization (MF). It offers several functional tools for gene expression matrix generation, expression matrix factorization, and metagene-based exploratory analysis including sample clustering, signature gene identification, and functional gene discovery. It should be noted that the metagenes in PCA can be ranked by the extent to which they explain the variation in the data, the features in both ICA and NMF are assumed to have equal weight.

Therefore, we noticed the user that the choice of MF techniques should be based on the purpose of one study with a preferred choice: PCA finds sources of separation in the data, which resulting in identification of dominated variation; ICA learns variation that are statically independent, which resulting in more accurate literature-derived association; NMF is well suited to transcriptional data, which is typically non-negative itself.

- easyMF project is hosted on <https://github.com/cma2015/easyMF>.
- easyMF docker image is available in <https://hub.docker.com/r/malab/easymf>.
- easyMF demo server can be accessed via <http://easymf.omicstudio.cloud>.
- The following part shows installation of easyMF docker image and detailed documentation for each function in easyMF.

## easyMF installation

---

- **Step 1:** Docker installation

### i) Docker installation and start ([Official installation tutorial](#))

For **Windows (Only available for Windows 10 Prefessional and Enterprise version):**

- Download [Docker](#) for windows;
- Double click the EXE file to open it;
- Follow the wizard instruction and complete installation;
- Search docker, select **Docker for Windows** in the search results and click it.

For **Mac OS X (Test on macOS Sierra version 10.12.6 and macOS High Sierra version 10.13.3):**

- Download [Docker](#) for Mac OS;
- Double click the DMG file to open it;
- Drag the docker into Applications and complete installation;
- Start docker from Launchpad by click it.

For **Ubuntu (Test on Ubuntu 18.04 LTS):**

- Go to [Docker](#), choose your Ubuntu version, browse to **pool/stable** and choose **amd64, armhf, ppc64el or s390x**. Download the **DEB** file for the Docker version you want to install;
- Install Docker, supposing that the DEB file is download into following path: `"/home/docker-ce~ubuntu_amd64.deb"`

```
$ sudo dpkg -i /home/docker-ce<version-XXX>-ubuntu_amd64.deb
$ sudo apt-get install -f
```

## ii) Verify if Docker is installed correctly

Once Docker installation is completed, we can run `hello-world` image to verify if Docker is installed correctly. Open terminal in Mac OS X and Linux operating system and open CMD for Windows operating system, then type the following command:

```
$ docker run hello-world
```

**Note:** root permission is required for Linux operating system.

- Once Docker is installed successfully, you will see the following message:

```
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:4fe721ccc2e8dc7362278a29dc660d833570ec2682f4e4194f4ee23e415e1064
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

```
To generate this message, Docker took the following steps:
```

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

```
To try something more ambitious, you can run an Ubuntu container with:
```

```
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
```

```
For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

- **Step 2:** easyMF installation from Docker Hub

```
# pull latest easyMF Docker image from docker hub
$ docker pull malab/easymf
```

- **Step 3:** Launch easyMF local server

```
$ docker run -it -p 8080:8080 malab/easymf bash
$ bash /home/galaxy/run.sh
```

Then, easyMF local server can be accessed via <http://localhost:8080>

Galaxy

Analyze Data Workflow Shared Data Visualization Help Login or Register

Using 8.0 GB

Tools

search tools

Get Data

easyMF

Workflows

All workflows

Welcome to easyMF

Matrix Factorization-based Transcriptome Analysis

Tutorial Test Data Docker Github

History

search datasets

Unnamed history

20 deleted

8 GB

This history is empty. You can load your own data or get data from an external source

## About

- easyMF is a user-friendly web platform that aims to facilitate biological discovery from large-scale transcriptome data through matrix factorization.
- easyMF comprises three functional modules, named **Matrix Preparation**, **Matrix Factorization**, and **Deep Mining**. Deep Mining includes Metagene-based Deep Mining Using AM and Metagene-based Deep Mining Using PM.
- easyMF is a flexible platform that can be used to perform accessible, reproducible, collaborative and transparent analyses of large-scale transcriptome data.
- easyMF was powered with an advanced packaging technology, which enables compatibility and portability.
- easyMF project is hosted on <https://github.com/cma2015/easyMF>, easyMF docker image is available at <https://hub.docker.com/r/malab/easymf>, easyMF server can be accessed via <http://easymf.omicstudio.cloud>.

1 Matrix Preparation

The diagram illustrates the Matrix Preparation workflow. It starts with a 'User' who performs an 'Upload' and 'Fetch' operation. The data is then processed through 'Data preprocess' to generate a 'Gene expression' matrix. This matrix is then used to create a 'Samples' matrix, which is finally used to generate a 'Gene expression' matrix.