easyMF User Mannual

(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
```

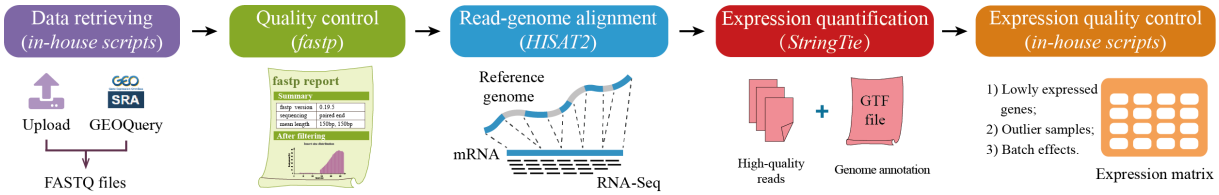
Then, easyMF local server can be accessed via http://localhost:8080


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
User
Upload
Fetch
Data preprocess
Samples
Genes
Gene expression

0. Matrix Preparation

Matrix factorization is typically started with an input of a gene expression matrix (genes in rows and individual samples in columns), which prompt us to design this module including three functions to prepare a high-quality gene expression matrix for downstream analysis.

The gene expression matrix can be automatically generated from raw reads using a bioinformatics pipeline (see following figure).



This module consists of three functions: **Data Retrieval**, **Expression Matrix Generation** and **Expression Quality Control**.

Functions/Tools	Description	Inputs	Outputs	Time (test data)	Program	References
Data Retrieval	Retrieve genome sequences, genome annotation, and RNA-Seq data automatically from public databases	Select a species; Database version; Data type	Genome sequences (in terms of Reference genome sequence); Genome annotation (in terms of Reference genome annotation); RNA-Seq data (in terms of Raw RNA-Seq data)	Depends on network speed	In-house scripts	This study
Expression Matrix Generation	Generate a gene expression matrix (genes in rows and individual samples in columns) through raw RNA-Seq quality control, read-genome alignment, and gene expression abundance calculation	Genome sequence and annotation; RNA-Seq data	Gene expression matrix	~ 2 mins	fastp (Raw RNA-Seq quality control)	Chen *et al.*, 2018
					HISAT2 (Read-genome alignment)	Kim *et al.*, 2015
					StringTie (Gene expression abundance calculation)	Pertea *et al.*, 2015
					In-house scripts	This study
Expression Quality Control	Generate a high-quality gene expression matrix through removing lowly expressed genes, outlier samples, or batch effects	Raw gene expression matrix	High-quality gene expression matrix	~ 10s	In-house scripts (Removing lowly expressed genes and outlier samples)	This study

Functions/Tools	Description	Inputs	Outputs	Time (test data)	Program	References
					sva (Removing batch effects)	Leek *et al.*, 2012

1. Data Retrieval

Data Retrieval can be used to retrieve **Genome sequences** and **Genome annotation** from Ensembl Plants, **RNA-Seq data** from NCBI (National Center for Biotechnology Information) GEO (Gene Expression Omnibus) or SRA (Short Read Archive) databases.

Inputs

For retrieving **genome sequences and annotation**, users need to select option **Obtain Genome Sequences and Annotation**.

- **Select a species**: This option provides the Latin name of 61 species.
- **Database version**: Ensembl releases from 25 to 47 are listed.
- **Data type**: Genome sequences (.fasta) or annotation (.gtf).

For retrieving **RNA-Seq** data, users need to select option **Obtain RNA-Seq data**.

- **Fecth data through data ID or ftp address**: easyMF provides two ways to download RNA-Seq data.

  If users select **Fetch data through data ID**, easyMF downloads RNA-Seq data by NCBI's tool *sratoolkit* (version 2.3.5) through RNA-Seq IDs (such as SRR1765337).

  If users select **Fetch data through data address**, easyMF downloads RNA-Seq data by wget using HTTP/FTP addresses.

Outputs

For **Obtain Genome Sequences and Annotation**

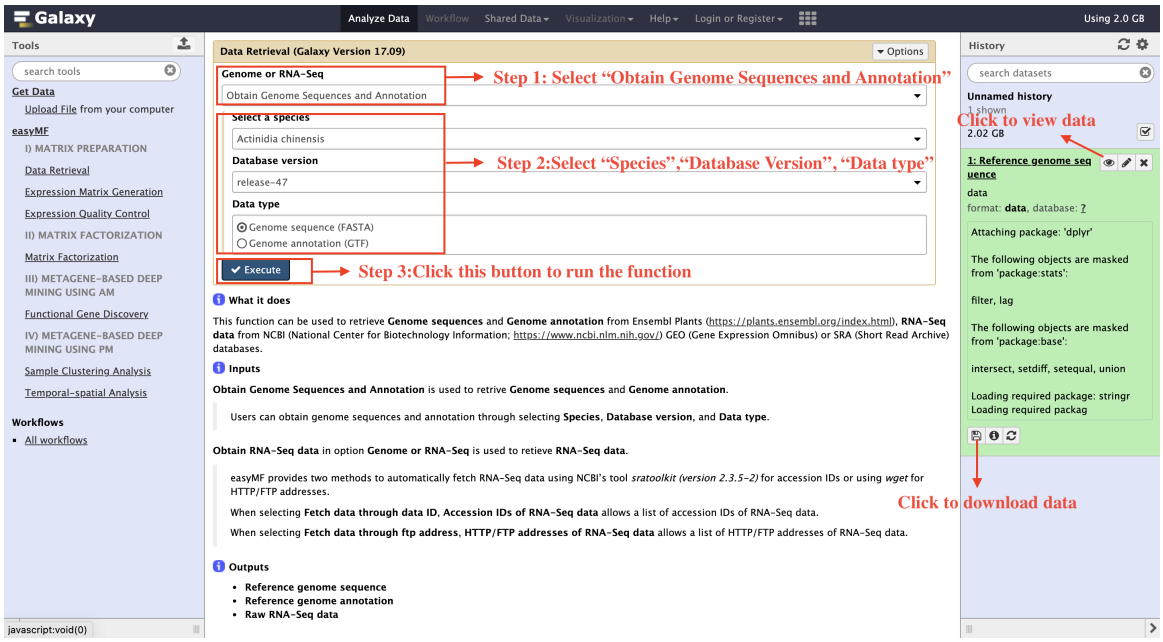
- **Reference genome sequence**
- **Reference genome annotation**

For **Obtain RNA-Seq data**

- **Raw RNA-Seq data**

How to use this function

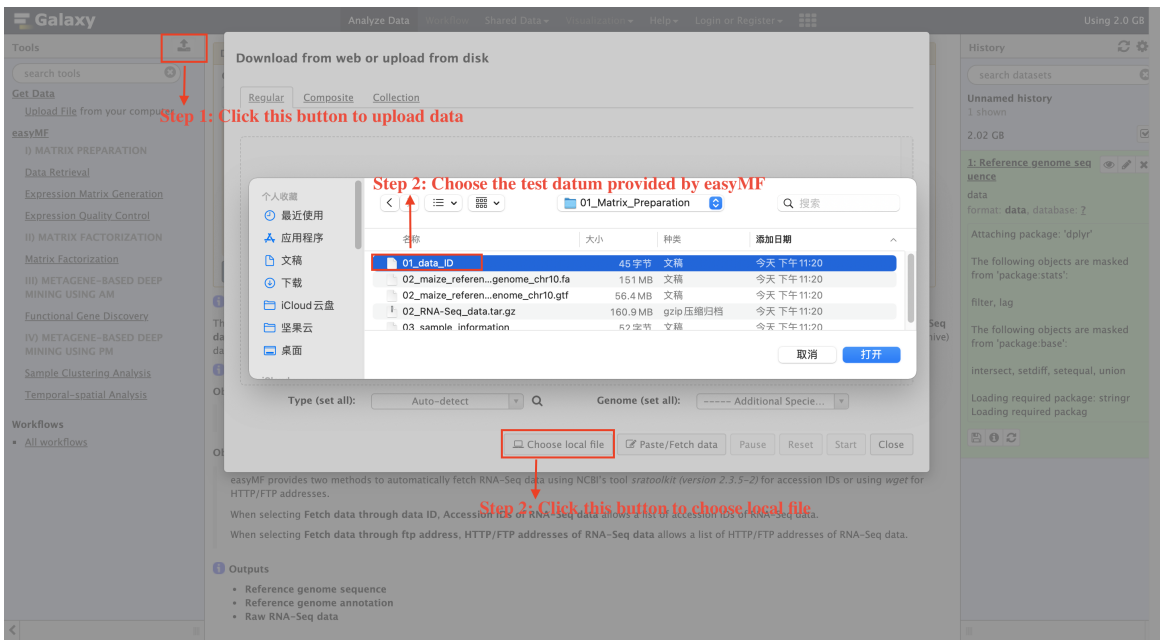
- The following screenshot shows us how to download genome sequences and annoatation using easyMF.

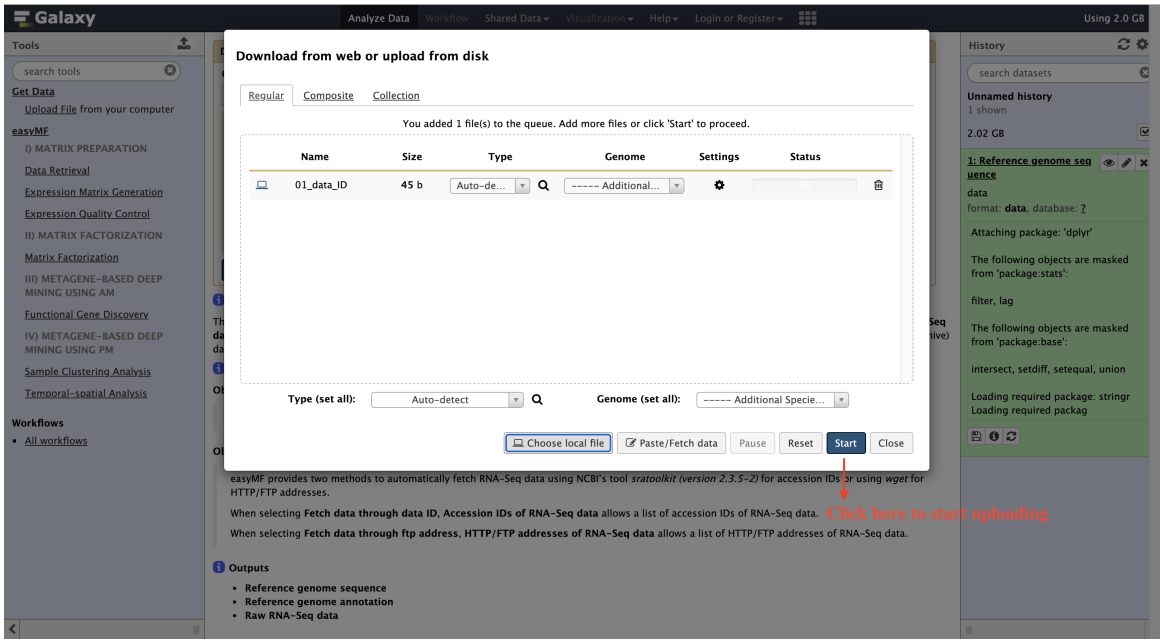


- The following screenshots show us how to download RNA-Seq data using easyMF.

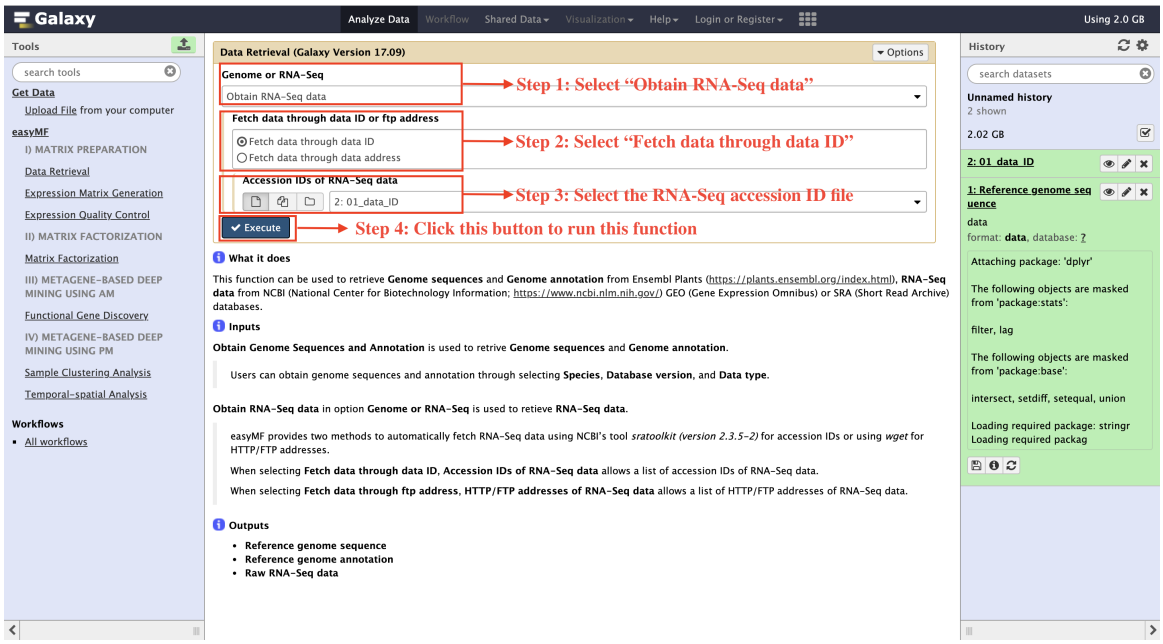
  Step 1: download test data provided by easyMF;

  Step 2: upload test datum `01_data_ID` in directory `Test_data/01_Matrix_Preparation` to history panel;





Step 3: input the corresponding file, and run the function.



Running time

Running time for the test data depends on network speed.

2. Expression Matrix Generation

This function can be used to generate a gene expression matrix (genes in rows and individual samples in columns) through raw RNA-Seq quality control, read-genome alignment, and gene expression abundance calculation.

Inputs

In **Data** section

- **Reference genome sequence**: Reference genome sequence in FASTA format used for read-genome alignment.
- **Reference genome annotation**: Reference genome annotation in GTF format used to estimate gene expression abundance.
- **Raw RNA-Seq data**: A compressed file containing RNA-Seq data in tar.gz format.

In **Parameters** section, easyMF needs users set parameters used for "RNA-Seq quality control" and "Read-genome alignment".

For "RNA-Seq quality control"

- **Minimum read length**: A threshold of read length that reads shorter than the length will be discarded.
- **The quality value that a base is qualified**: A threshold of base quality value to trim low-quality reads.

For "Read-genome alignment"

- **Minimum intron length for RNA-Seq alignment**
- **Maximum intron length for RNA-Seq alignment**

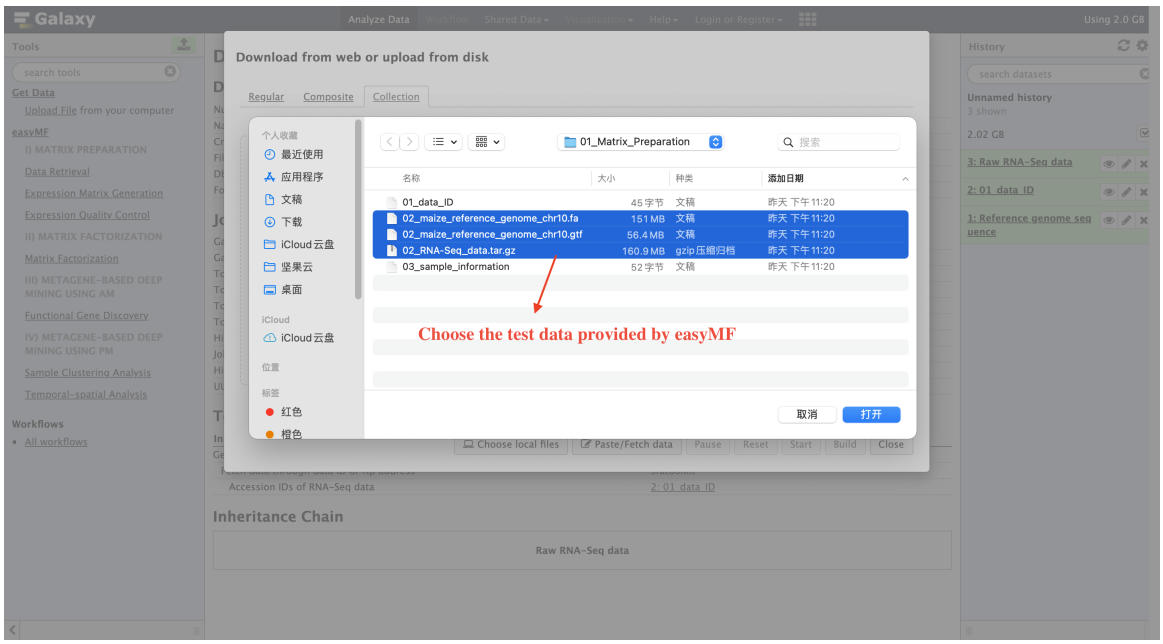
Output

- **Raw Gene Expression Matrix**: A gene expression matrix (genes in rows and individual samples in columns).

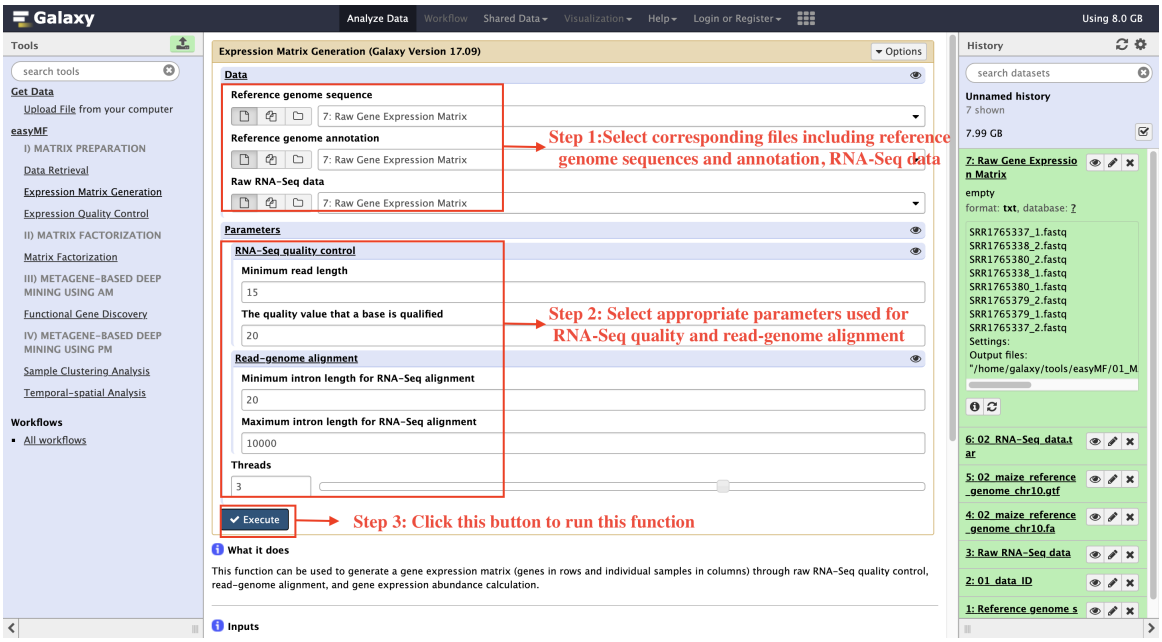
How to use this function

- Test data for this function are in directory `Test_data/01_Matrix_Preparation` including `02_maize_reference_genome_chr10.fa`, `02_maize_reference_genome_chr10.gtf`, and `02_RNA-Seq_data.tar.gz`.
- The following screenshots show us how to generate a gene expression matrix using easyMF.

  Step 1: upload test data in directory `Test_data/01_Matrix_Preparation` to history panel;



  Step 2: input the corresponding files and appropriate parameters, then run the function.



Running time

This step will cost ~ 2 mins for the test data.

3. Expression Quality Control

Once gene expression matrix generated, to accurately implement MF-based analysis, quality of the gene expression matrix need to be improved, which can be operated through three different dimensions including **removing lowly expressed genes**, **removing outlier samples**, **removing batch effects**.

Inputs

For **Removing lowly expressed genes**

- **Expression value of expressed genes**: Expression value of genes regarded as expressed.
- **Minimum sample number**: The number of samples of expressed genes.

easyMF provides default values for these two parameters: **Expression value of expressed genes** (default as 1) and **Minimum sample number** (default as 3), which means genes regarded as expressed with expression value greater than 1 in at least 3 samples.

For **Removing outlier samples**

- **Threshold of potential repeat samples**: Expression values between two samples are almost identical.
- **Threshold of low-quality samples**: Sample distance between two RNA-Seq data.

For **Removing batch effects**

- **Sample information**: RNA-Seq samples with batch information. In the text, the first column presents sample IDs, and the second column presents batch information distinguished with Arabic numerals.

```
SRR1765379  1
SRR1765380  1
SRR1765337  2
SRR1765338  2
```

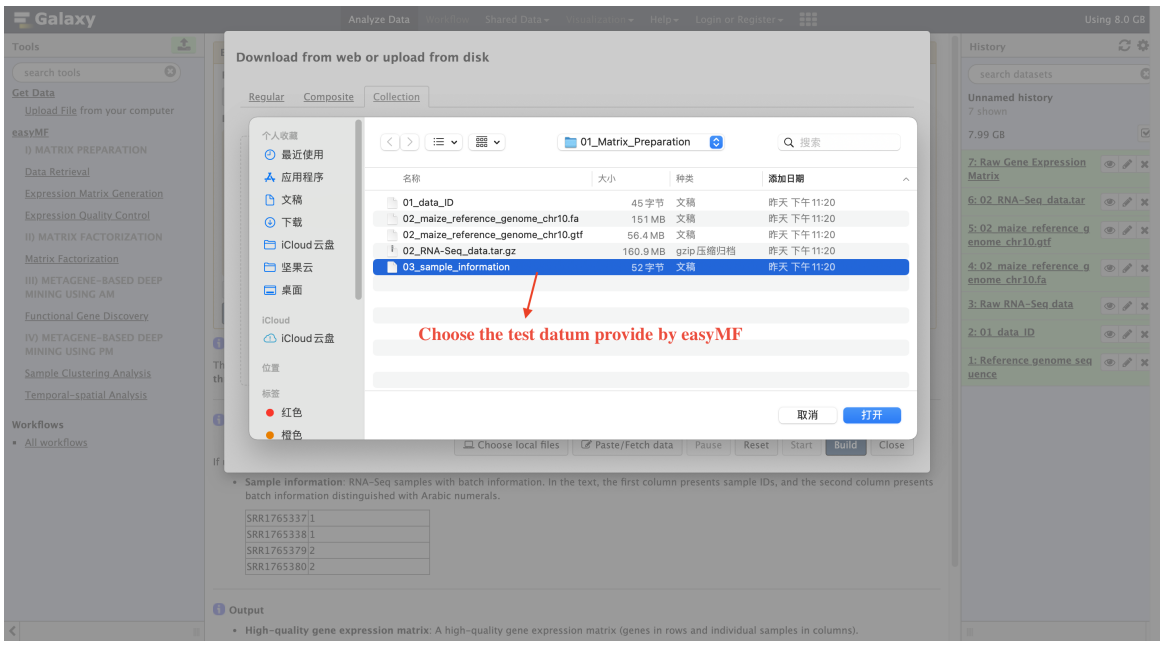
Output

- **High-quality gene expression matrix**: A high-quality gene expression matrix (genes in rows and individual samples in columns).

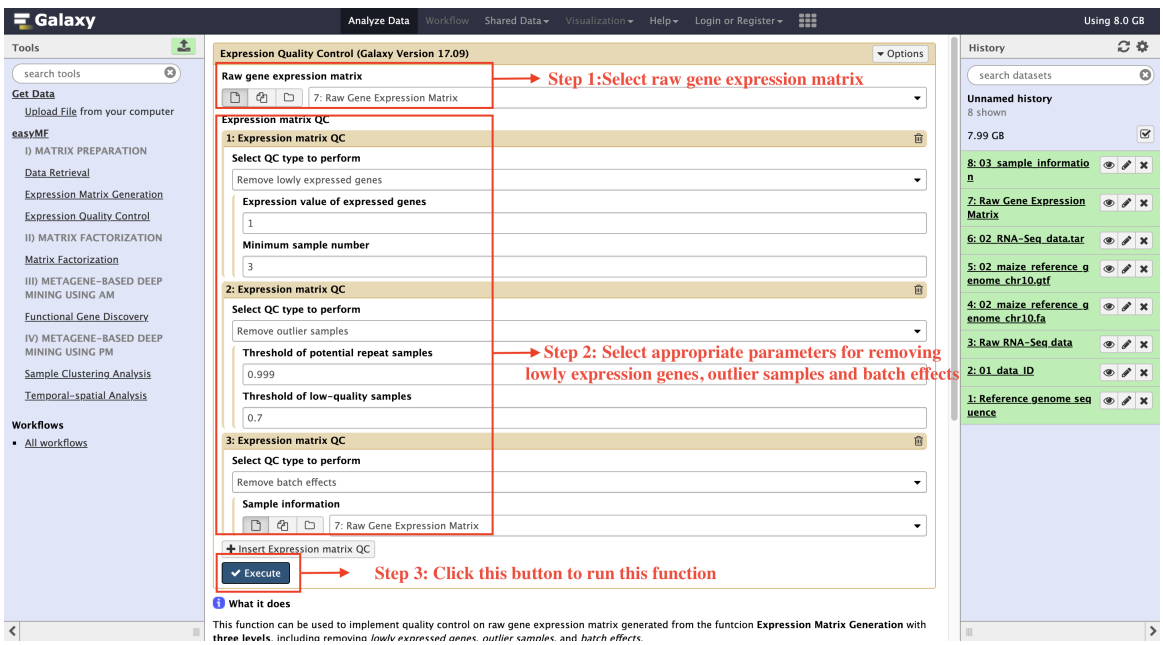
How to use this function

- Test data for this function are `03_sample_information` in `01_Matrix_Preparation` for **Sample information**, and **Raw gene expression matrix** generated by the function **Expression Matrix Generation**.
- The following screenshota show us how to generate a high-quality gene expression matrix using easyMF.

  Step 1: upload test datum in directory `Test_data/01_Matrix_Preparation` to history panel;



  Step 2: input the corresponding files and appropriate parameters, then run the function.



Running time

This step will cost ~ 10s for the test data.

0. Matrix Factorization

Matrix Factorization can be used to decompose a gene expression matrix into an amplitude matrix (AM) and a pattern matrix (PM) using three different algorithms, i.e., principal component analysis (PCA), independent component analysis (ICA), and non-negative matrix factorization (NMF).

Functions/Tools	Description	Inputs	Outputs	Time (test data)	Program	References
Matrix Factorization	Decompose a gene expression matrix into a product of an amplitude matrix (AM) and pattern matrix (PM)	Gene expression matrix	Amplitude matrix; Pattern matrix	~ 1 mins	prcomp (PCA)	This study
					ica (ICA)	Helwig, 2015
					bignmf (NMF)	Pan *et al.*, 2012

Inputs

- **Decomposition options**: Different algorithms used for gene expression matrix decomposition including PCA, ICA, and NMF.
- **Gene expression matrix**: A gene expression matrix (genes in rows and individual samples in columns).
- **Metagene number**: The number of metagenes decomposed from the gene expression matrix. In current version, easyMF provides three options for users to set this number:

1) **Scan an optimal metagene number automatically by easyMF**: easyMF can automatically chosen an appropriate number using two methods: internal consistency of Cronbach's α value for **PCA** and inflection point of the rate of the mean residual decline for **ICA** and **NMF**;

2) **Set each sample as a metagene**: Setting the number of samples in the gene expression matrix as the number of metagenes;

3) **Specify metagene number by users**: A specified metagene number user provided.

Outputs

- **Statistics analysis of the decomposition**: If users need easyMF automatically chosen an optimal metagene number, easyMF would provide a comprehensive report statistics analysis of the decomposition.
- **Amplitude matrix**: Amplitude matrix decomposed from gene expression matrix (genes in rows and metagenes in columns).

	Metagene 1	Metagene 2	...	Metagene n
Zm00001d053636	0.080	-0.889	...	2.029
Zm00001d053632	1.338	0.729	...	-0.049
...	...	...	...	...
Zm00001d053635	-1.674	0.036	...	-0.494

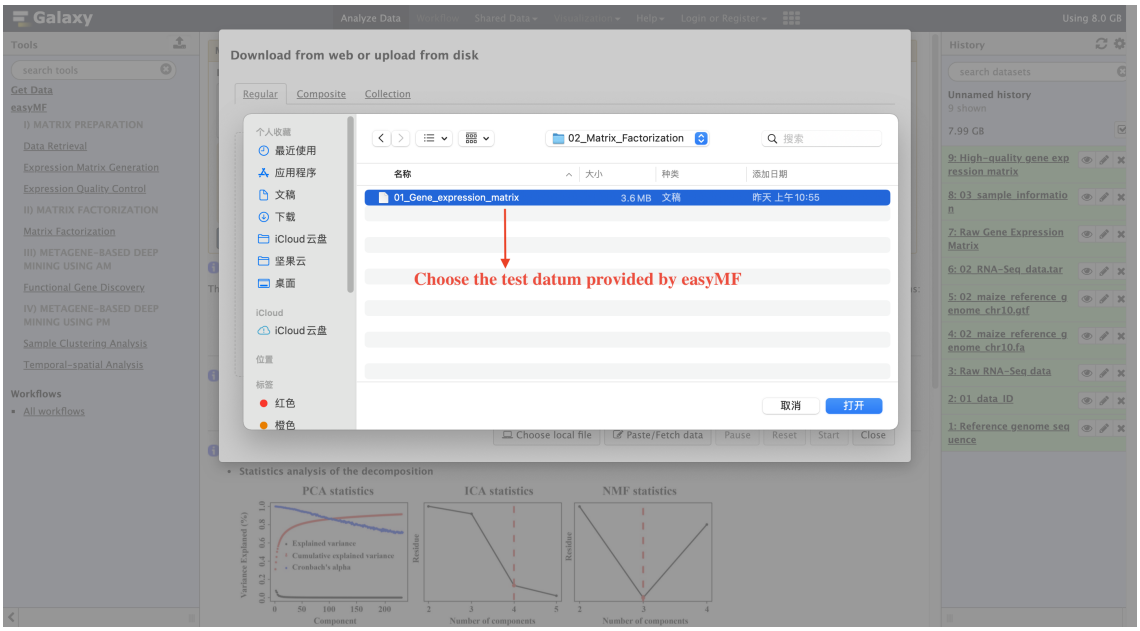
- **Pattern matrix**: Pattern matrix decomposed from gene expression matrix (samples in rows and metagenes in columns).

	Metagene 1	Metagene 2	...	Metagene n
Sample 1	-2.081	0.663	...	-0.711
Sample 2	-2.114	0.711	...	-0.757
...	...	...	...	...
Sample 4	-2.185	0.671	...	-0.719

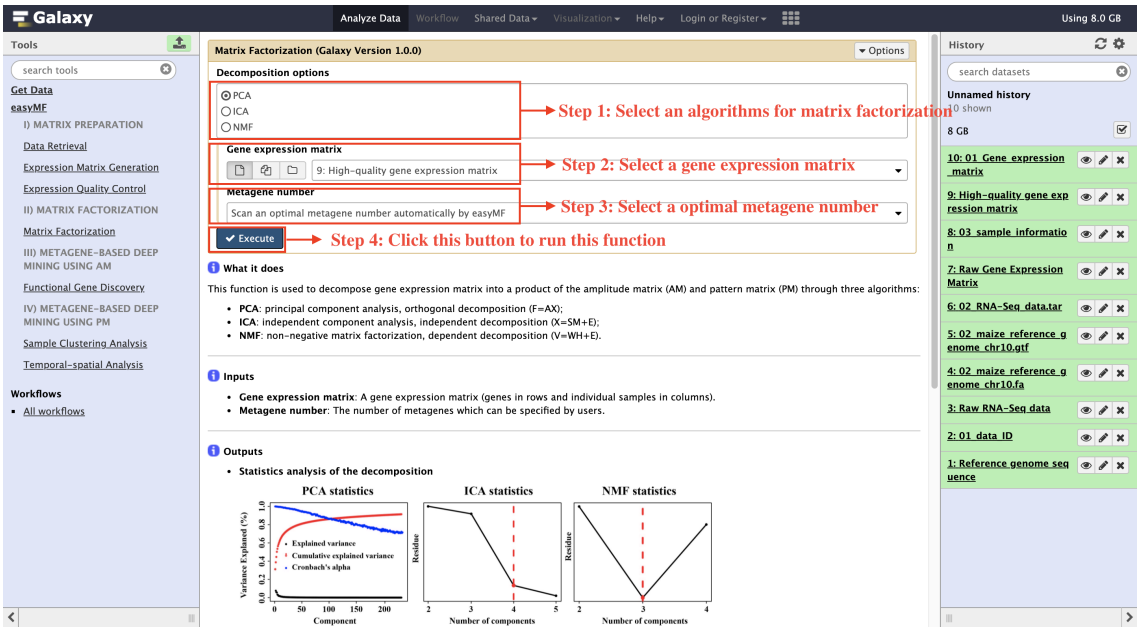
How to use this function

- Test datum for this function is `01_Gene_expression_matrix` in directory `Test_data/02_Matrix_Factorization`.
- The following screenshots show us how to decompose the gene expression matrix using easyMF.

  Step 1: upload test data in directory `Test_data/02_Matrix_Factorization` to history panel;



  Step 2: input the corresponding files and appropriate parameters, then run the function.



Running time

This step will cost ~ 1 mins for the test data.

0. Metagene-based Deep Mining Using AM

Amplitude matrix (AM), a matrix with genes in rows and metagenes in columns, describes gene-level relationships. In current version of easyMF, users can make use of AM for functional gene discovery.

This module consists of one function: **Functional Gene Discovery**.

Functions/Tools	Description	Inputs	Outputs	Time (test data)	Program	References
Functional Gene Discovery	Calculate gene score and rank genes based on the probability of their association with a specific biology function	Amplitude matrix; A set of genes with a specific characteristic	Gene score and rank; Area under the self-ranked curve (AUSR) plot	~ 10s	In-house scripts	Fehrmann *et al.*, 2015

1. Functional Gene Discovery

Functional gene discovery can be used to calculate gene score and rank genes based on the probability of their association with a specific biology function.

Inputs

- **Amplitude matrix**: An amplitude matrix of AM coefficients with genes in rows and metagenes in columns. Here is an example:

	Metagene 1	Metagene 2	Metagene 3	...	Metagene n
Zm00001d053636	0.080	-0.889	1.504	...	2.029
Zm00001d053632	1.338	0.729	-0.113	...	-0.049
...	...	...	...	...	...
Zm00001d053635	-1.674	0.036	-0.047	...	-0.494

- **Functional genes**: A set of genes associated with a specific biology function, such as enriched in a phenotype of interest. If users select **Upload a file with functional gene IDs from local**

disk, a newline-delimited file containing gene IDs needs to be provided; if users select **Enter functional gene IDs**, gene IDs need to be separated by comma. Here are two examples:

A newline-delimited file containing gene IDs for **Upload a file with functional gene IDs from local disk**:

```
Zm00001d053636
Zm00001d053632
Zm00001d053630
...
Zm00001d053635
```

Comma-separated gene IDs for **Enter functional gene IDs**:

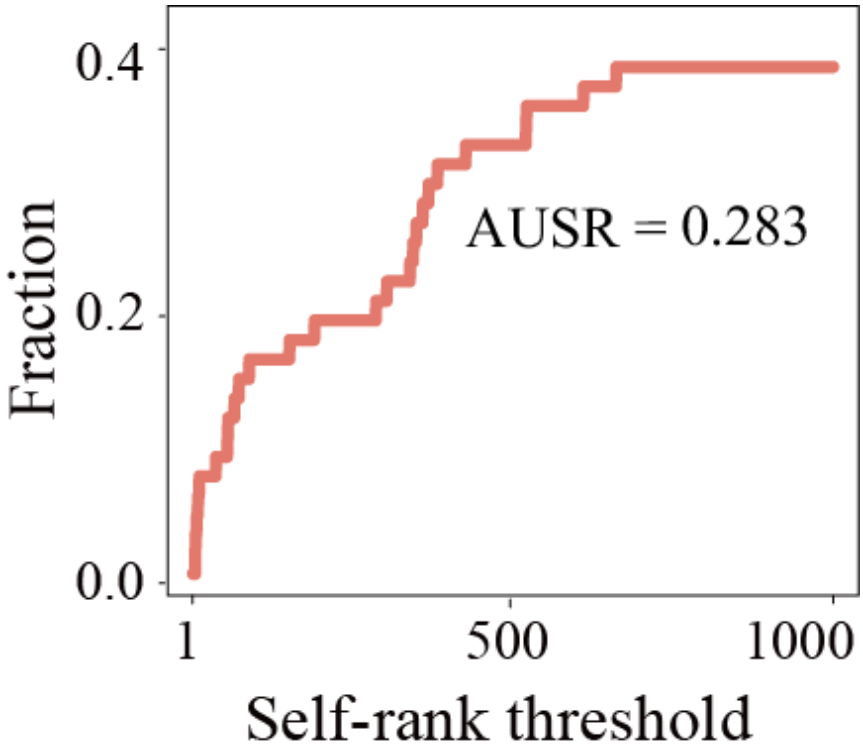
```
Zm00001d053636,Zm00001d053632,Zm00001d053630,...,Zm00001d053635
```

Outputs

- **Gene score and rank**: Summary of gene prioritization results containing **Gene ID**, **Score**, **Rank**, and **Annotation**. The higher ranking of a gene, the more related to the biological function. Here is an example:

Gene ID	Score	Rank	Annotation
Zm00001d053636	1	1	Label
Zm00001d053632	0.888	3	Label
Zm00001d004839	1	1	Unlabel
...	...	...	...
Zm00001d053635	0.92	2	Unlabel

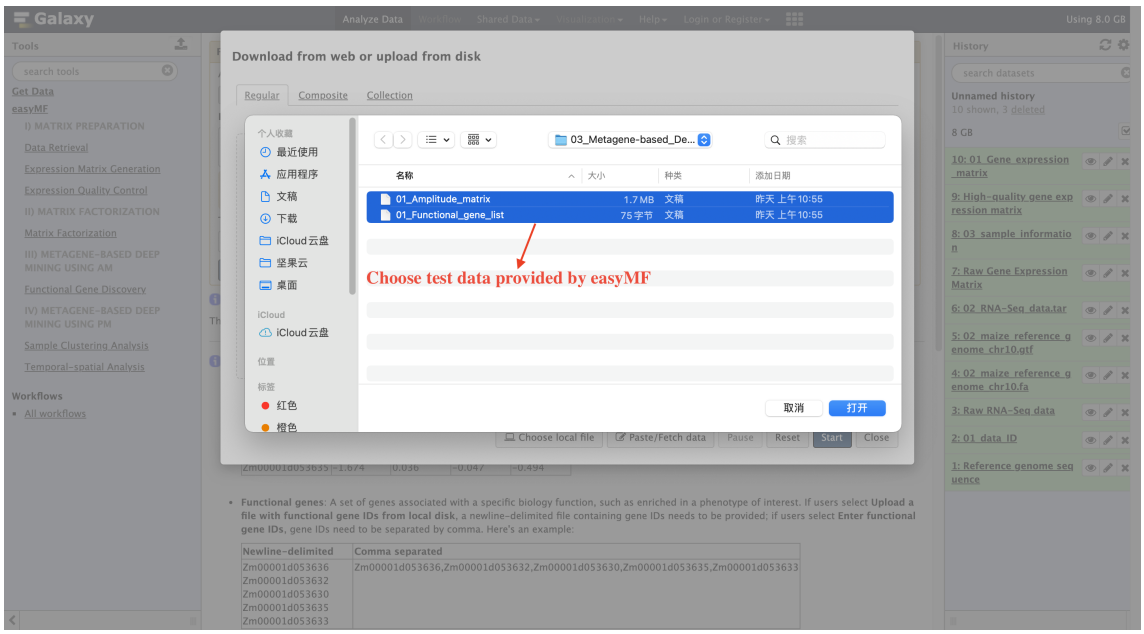
- **Area under the self-ranked curve (AUSR) plot**: A plot of ratio (*Ra*) along the *y* axis versus self-rank along the *x* axis, where rank represents the ranks of all positive genes, *Ra*(*l*) represents the ratio of ranks lower than a pre-defined level of *l*.



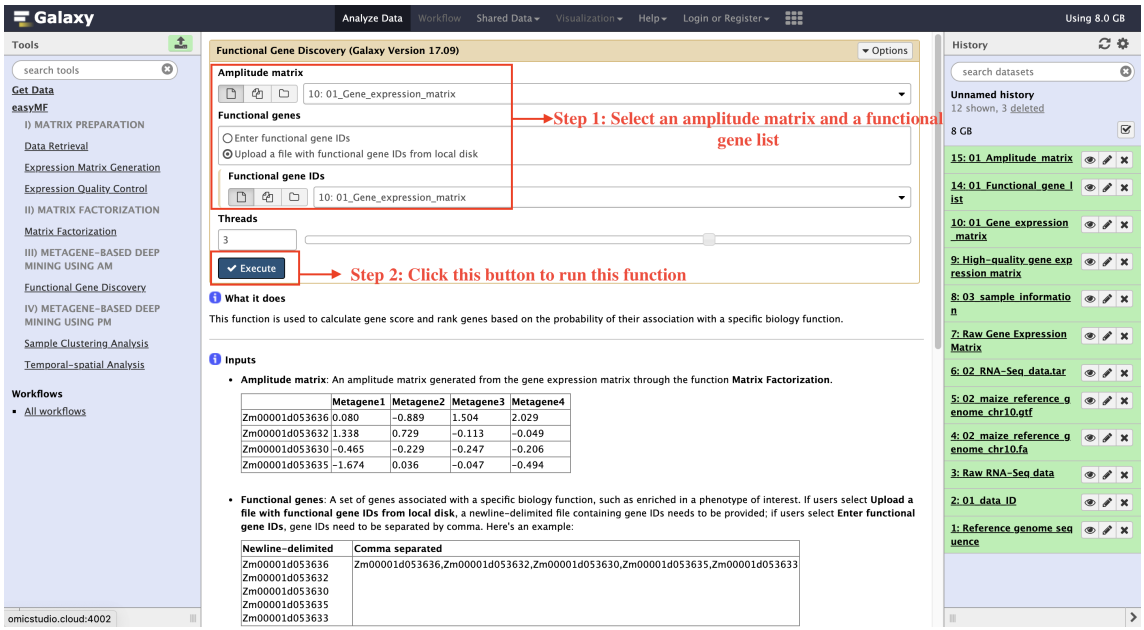
How to use this function

- Test data for this function are in directory `Test_data/03_Metagene-based_Deep_Mining_Using_AM` including.
- The following screenshots show us how to implement functional gene discovery using easyMF.

  Step 1: upload test data in directory `Test_data/03_Metagene-based_Deep_Mining_Using_AM` to history panel;



  Step 2: input the corresponding files and appropriate parameters, then run the function.



Running time

This step will cost ~ 10s for the test data.

0. Metagene-based Deep Mining Using PM

Pattern matrix (AM), a matrix with samples in rows and metagenes in columns , describes sample-level relationships. In current version of easyMF, users can make use of PM for sample clustering, temporal and spatial transcriptome analysis.

This module consists of two functions: **Sample Clustering Analysis**, and **Temporal-spatial Analysis**.

Functions/Tools	Description	Inputs	Outputs	Time (test data)	Program	References
Sample Clustering Analysis	Cluter samples automatically based on the pattern matrix	Pattern matrix	Cluster information; Cluster visualization;	~ 15s	mclust	Scrucca *et al.*, 2016
					apcluster	Bodenhofer *et al.*, 2011
					SSE	This study
					fpc	Hennig, 2013
					vegan	Dixon, 2003
					gap	Maechler *et al.*, 2012
Temporal-spatial Analysis	Detect functional modules and identity signature genes	Gene expression matrix; Amplitude matrix; Pattern matrix; Sample information	Signature genes of each module; GO enrichment analysis of each module; Visualization of specific module	~ 5 mins	In-house scripts	This study
					cogaps	Stein-O'Brien *et al.*, 2017
					topGO	Alexa and Rahnenführer, 2009

1. Sample Clustering Analysis

In the current version, easyMF provides six optional algorithms (mclust, apcluster, SSE, fpc, vegan, and gap) to cluster samples using PM coefficients. The cluster result is visualized in dot plots and tables, providing a quick overview of the relationships among samples.

Inputs

- **Pattern matrix**: A pattern matrix with samples in rows and metagenes in columns. Here is an example:

	Metagene 1	Metagene 2	...	Metagene 4
Sample 1	-2.081	0.663	...	-0.711
Sample 2	-2.114	0.711	...	-0.757
...	...	...	...	...
Sample 4	-2.185	0.671	...	-0.719

- **Cluster algorithms**: easyMF provides six cluster algorithms to be cluster samples including mclust, apcluster, SSE, fpc, vegan, and gap.

  mclust: mclust implements model-based clustering using parameterized finite Gaussian mixture models. Models are estimated by Expectation Maximization (EM) algorithm initialized through hierarchical model-based agglomerative clustering. The optimal model is then selected according to Bayesian information criterion (BIC).

  apcluster: affinity propagation clusters data using a set of real-valued pairwise data point similarities as input. It iterates and searches for clusters maximizing an objective net similarity function.

  SSE: SSE calculates sum of square error based on k-means algorithm, and selects the best number of clusters based on inflection point of the residuals.

  fpc: fpc optimums average silhouette width, partitioning around medoids with estimation of cluster number.

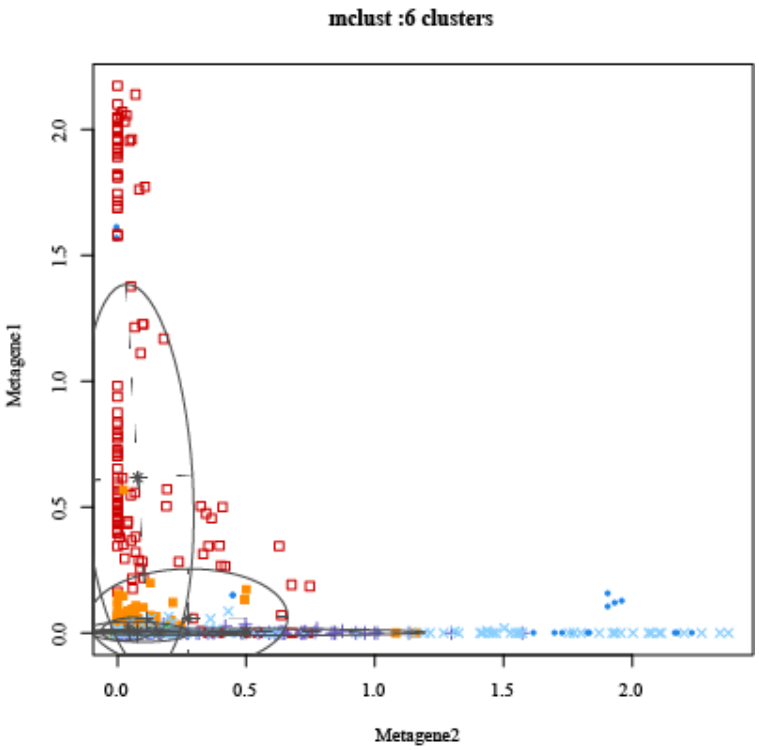
  vegan: vegan is a cascade k-means partitioning using a range of *k* values. It generates the optimal cluster based on Calinski-Harabasz criterion.

  gap: gap calculates a goodness of clustering measure of the gap statistic for estimating the number of clusters.

  ```
  NOTE: easyMF supports multi-selection for different algorithms.
  ```

Outputs

- **cluster visulization**: A dot plot of the clustering results.



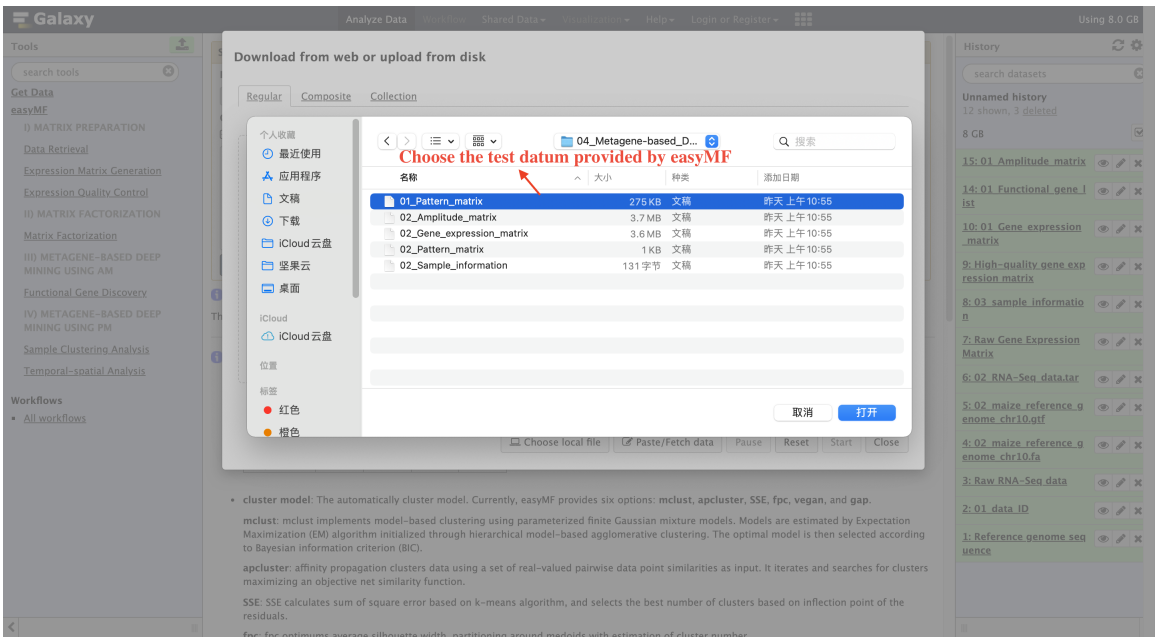
- **cluster information**: Sample cluster results for specific algorithms.

	mclust	apcluster
Sample1	1	18
Sample7	7	18
Sample11	3	21
Sample15	7	14

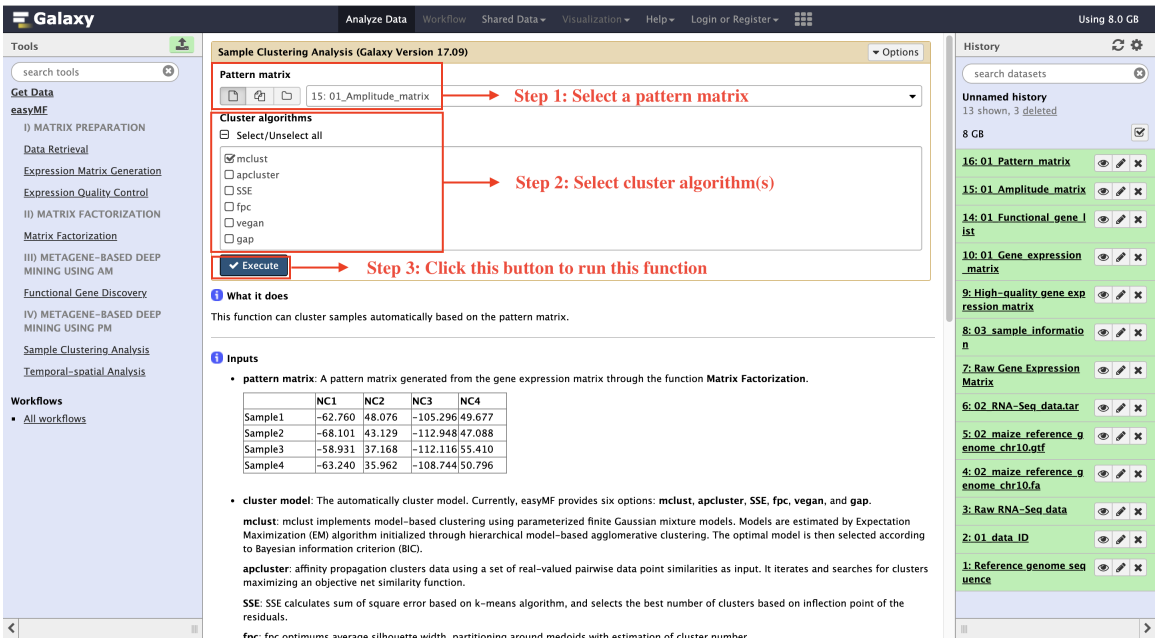
How to use this function

- Test datum for this function is `01_Pattern_matrix` in directory `Test_data/04_Metagene-based_Deep_Mining_Using_PM`.
- The following screenshots show us how to cluster samples using easyMF.

  Step 1: upload test datum in directory `Test_data/01_Matrix_Preparation` to history panel;



  Step 2: input the corresponding files and appropriate parameters, then run the function.



Running time

This step will cost ~ 15s for the test data.

2. Temporal-spatial Analysis

easyMF can be used to determine the extent to which genes change over time in response to perturbations (e.g., developmental time) and identify signature genes dominated at specific compartments with spatial resolution in individual tissue samples (spatial transcriptomes).

Inputs

In **Data** section

- **Gene expression matrix**: A gene expression matrix generated by the module **Matrix Preparation**.
- **Amplitude matrix**: An amplitude matrix with genes in rows and metagenes in columns decomposed by the input gene expression matrix .
- **Pattern matrix**: A pattern matrix with samples in rows and metagenes in columns decomposed by the input gene expression matrix .
- **Sample information**: Sample information containing development stages or spatial compartments.

In **Parameters** section

- **Threshold of Pearson Correlation Coefficient**: A Pearson Correlation Coefficient value used for signature gene identification.
- **Threshold of P-value**: A *P*-value used for signature gene identification.
- **Select a species**: Species name which can be selected from drop-down menu is used to annotate gene information.

Outputs

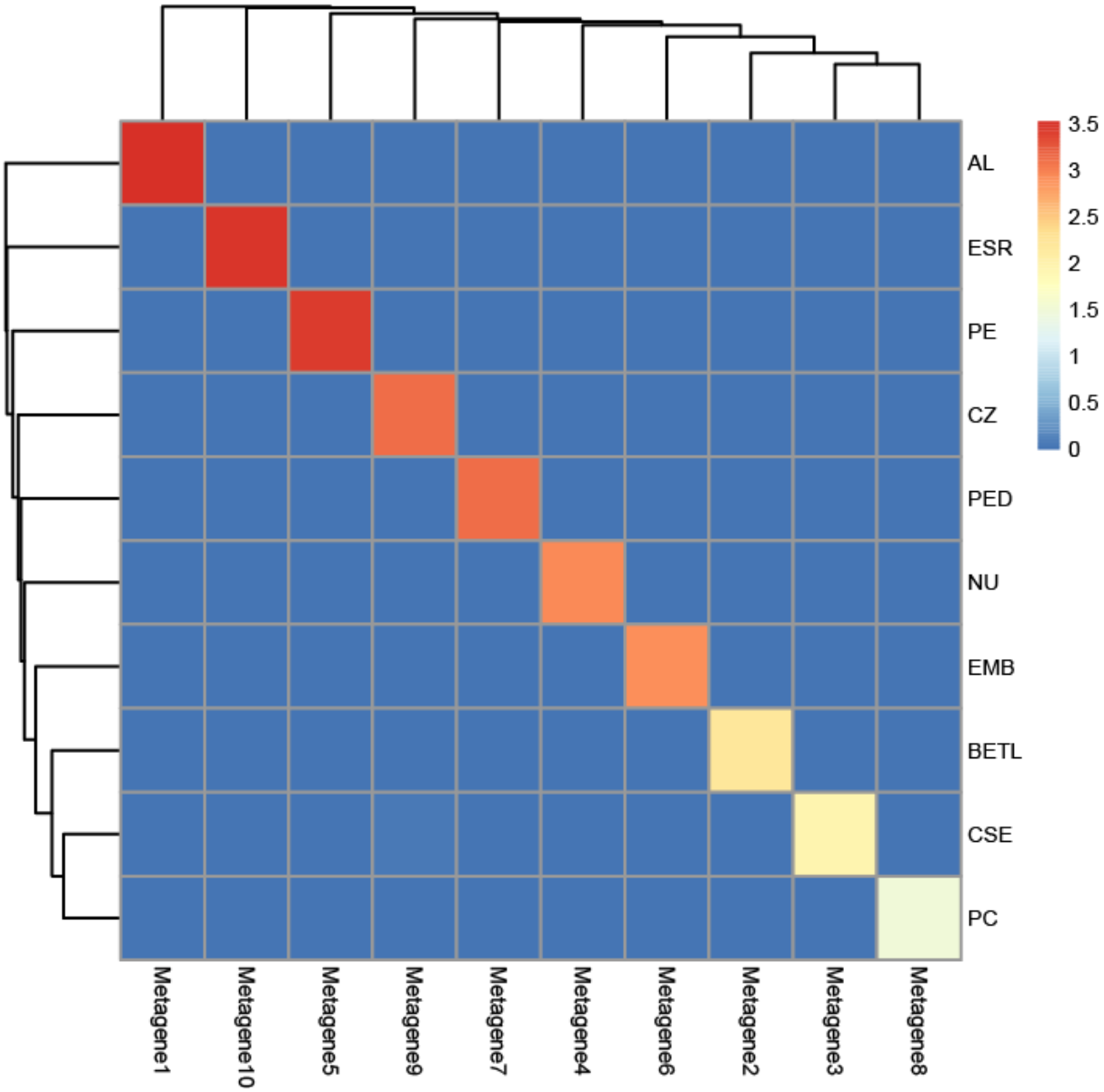
- **Signature genes of each metagene**: Summary of signature genes in each metagene. Columns represent **Metagene ID**, **Gene ID**, and **Gene description**, respectively.

Metagene ID	Gene ID	Gene description
Metagene1	Zm00001d020505	DNA glycosylase superfamily protein
Metagene2	Zm00001d012083	Thioredoxin F-type chloroplastic
Metagene3	Zm00001d033585	Leaf permease1

- **GO enrichment analysis of each metagene**: Summary of GO enrichment results of signature genes in each metagene.

Metagene ID	Type	GO	Term	Annotated	Significant	Expected	e
Metagene1	BP	GO:0042445	hormone metabolic process	91	6	1.65	0

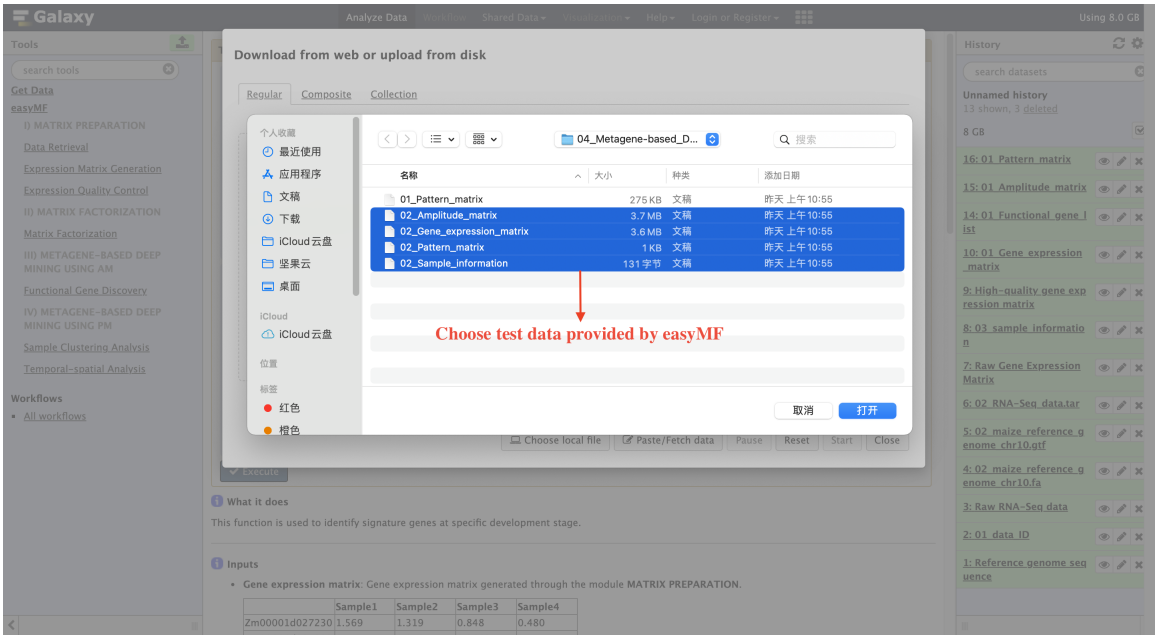
- **Visualization of metagenes**: Hierarchical clustering analysis of pattern matrix.



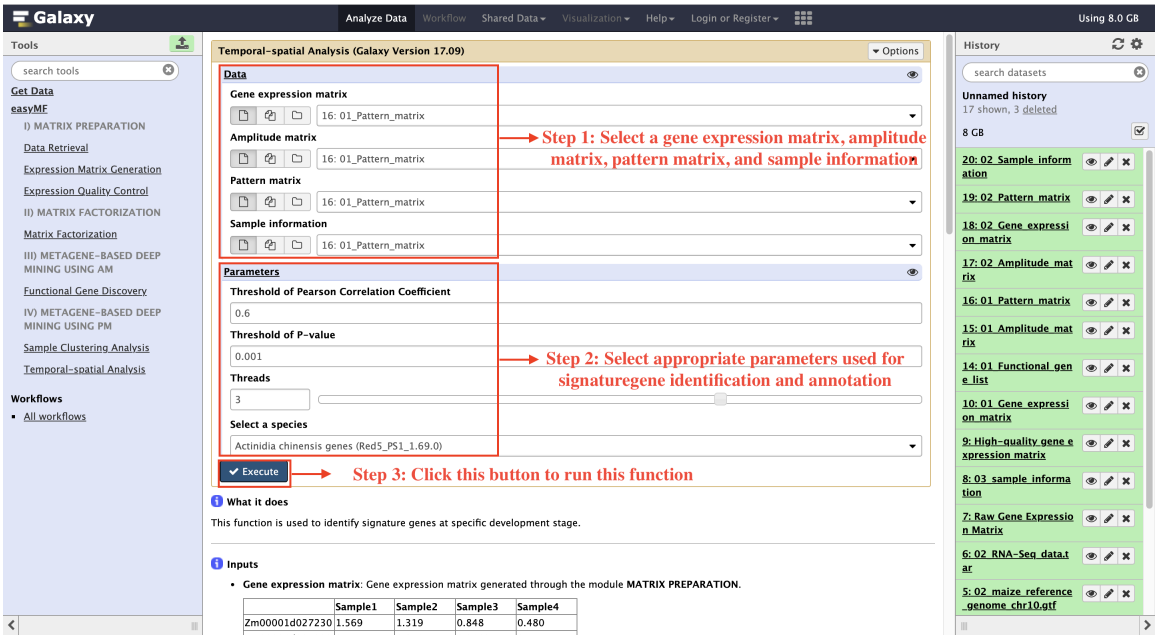
How to use this function

- Test data for this function are in directory `Test_data/04_Metagene-based_Deep_Mining_Using_PM` including `02_Gene_expression_matrix`, `02_Amplitude_matrix`, `01_Pattern_matrix`, and `02_Sample_information`.
- The following screenshots show us how to implement temporal-spatial transcriptome analysis using easMF.

  Step 1: upload test data in directory `Test_data/04_Metagene-based_Deep_Mining_Using_PM` to history panel;



Step 2: input the corresponding files and appropriate parameters, then run the function.



Running time

This step will cost ~ 5 mins for the test data.