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.

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.

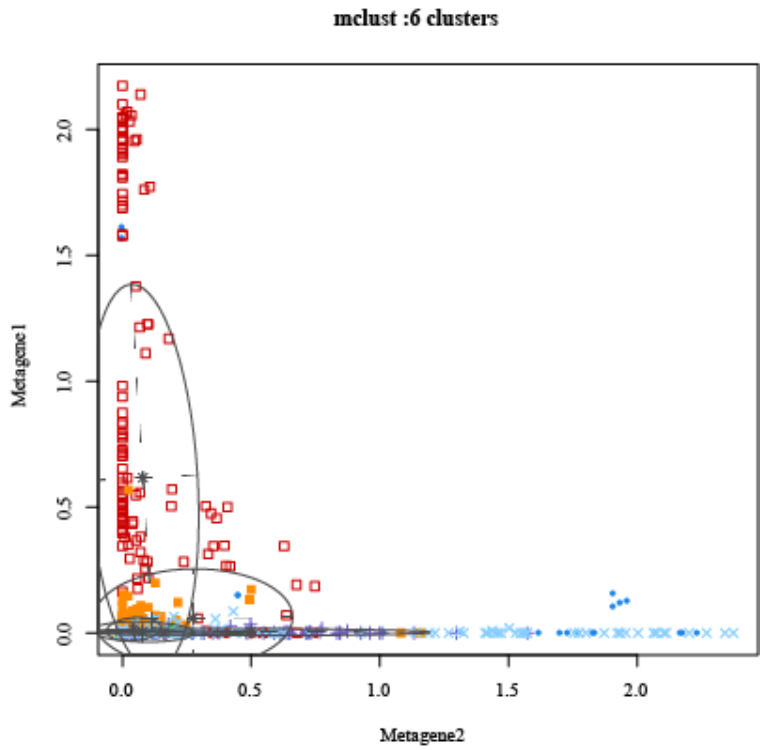
vegan: vegan is a cascade k-means partitioning using a range of *k* values. It generates the optimal cluster based on Calinski-Harabasz criterion.

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.



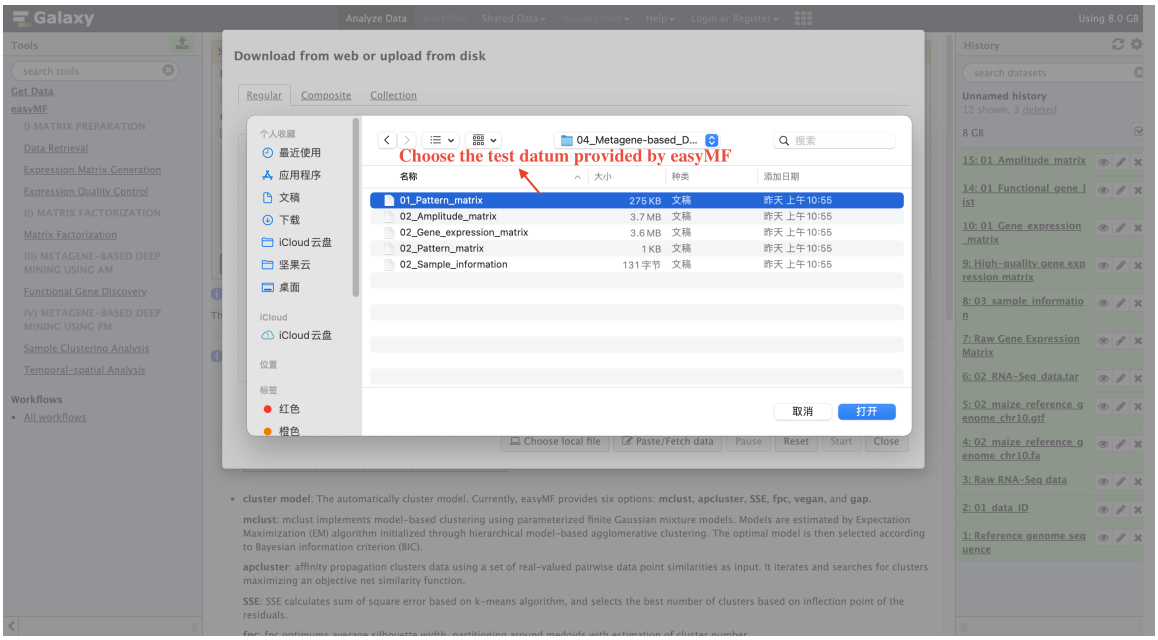
- **cluster information**: Sample cluster results for specific algorithms.

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

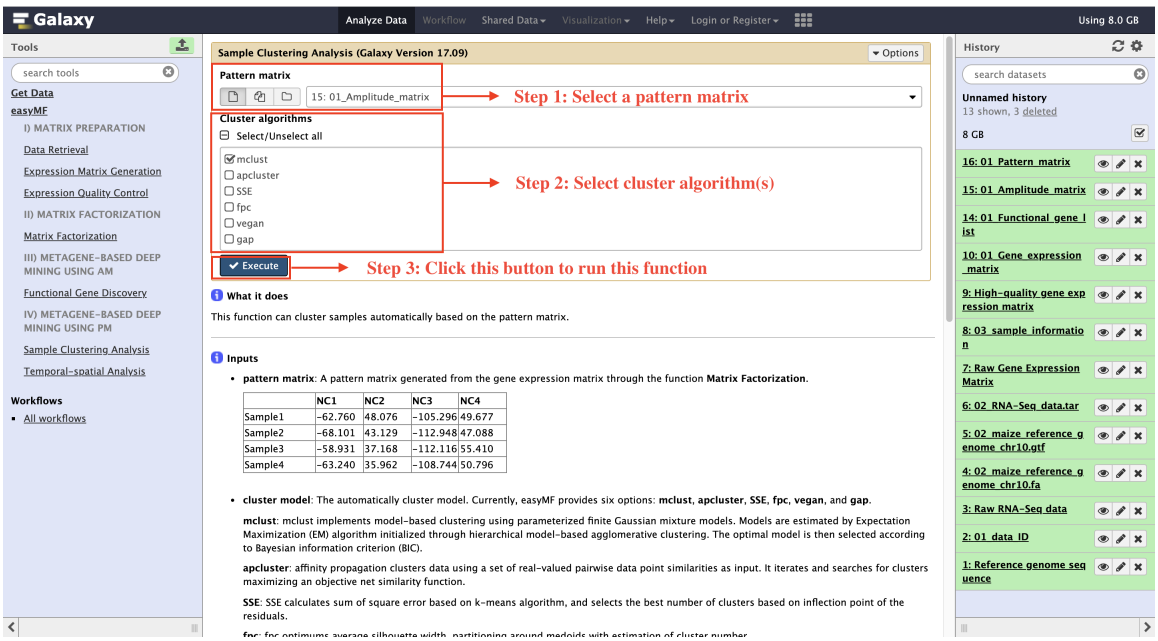
How to use this function

- 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

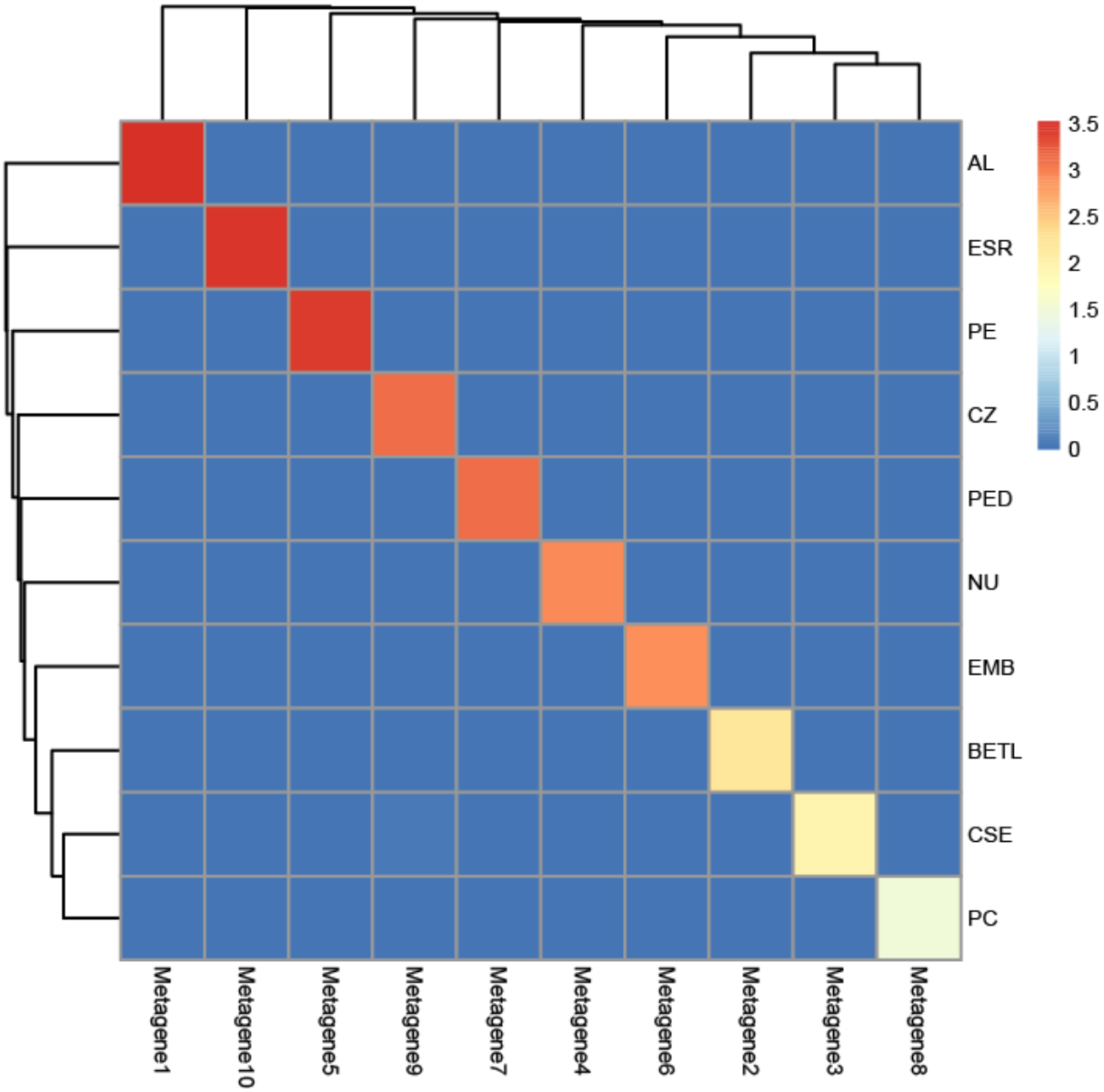
- **Signature genes of each metagene**: Summary of signature genes in each metagene. Columns represent **Metagene ID**, **Gene ID**, and **Gene description**, respectively.

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

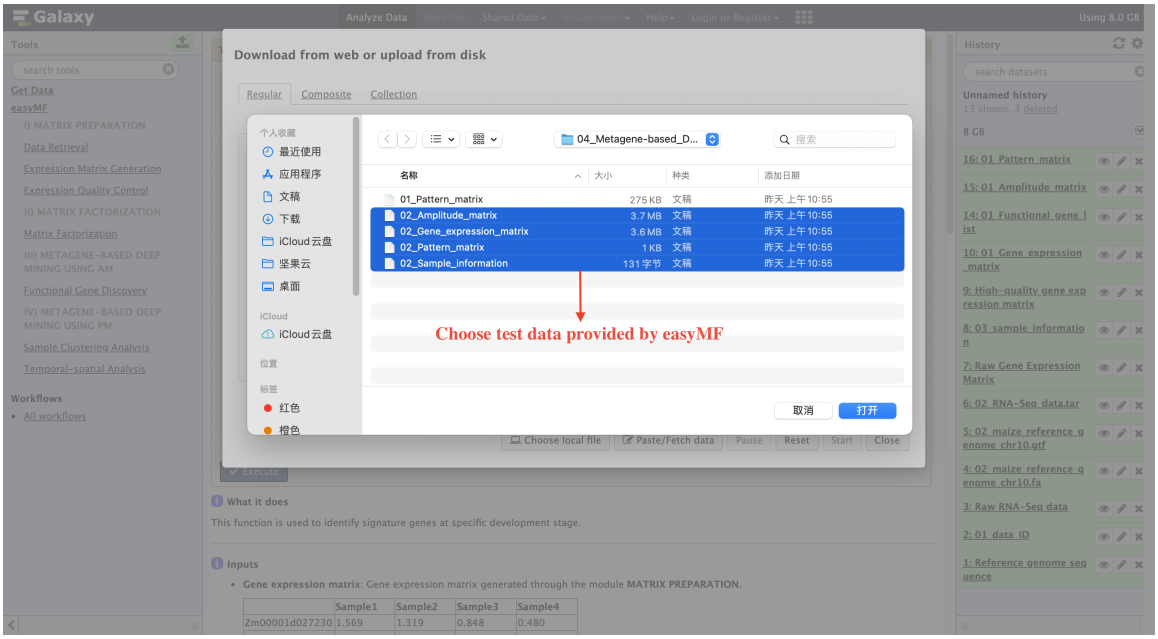
- **Visualization of metagenes**: Hierarchical clustering analysis of pattern matrix.



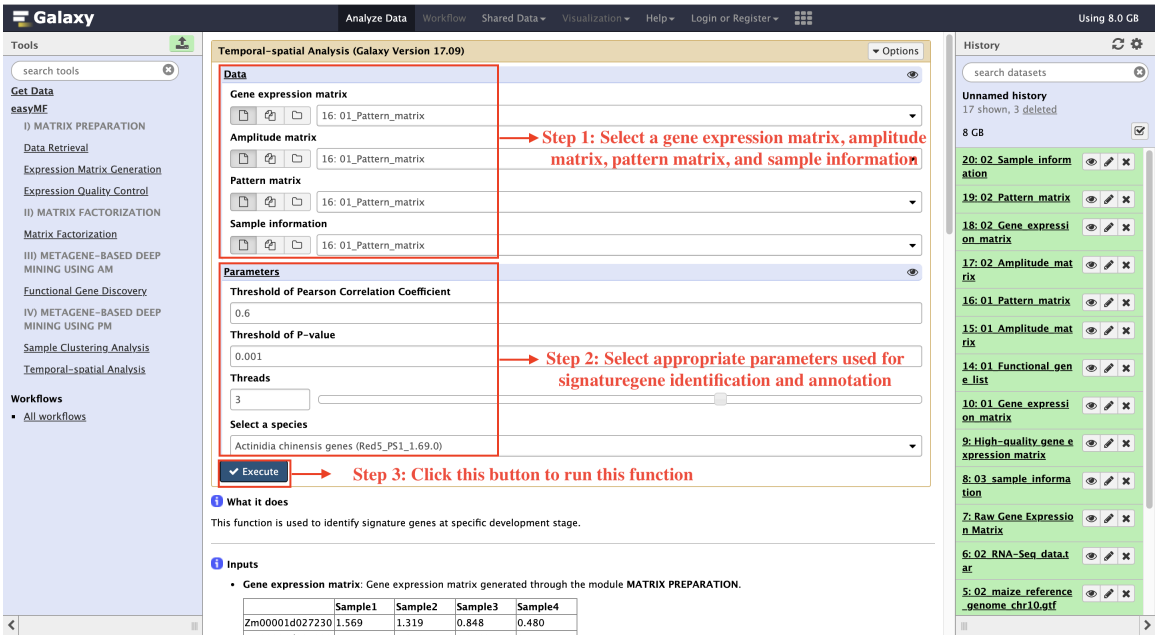
How to use this function

- 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.