

# 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, functional gene discovery, subtype cell detection, and pathway activity inference.

- 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 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 and pathway activity inference.

This module consists of two functions: **Functional Gene Discovery**, and **Pathway Activity Inference**.

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	<a href="#">Fehrmann et al., 2015</a>
Pathway Activity Inference	Examine the pathway activity for any gene set of interest	Amplitude matrix; Pathway annotation; A set of genes with a specific characteristic	Activated pathways	~ 1 mins	In-house scripts	This study

# 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**:

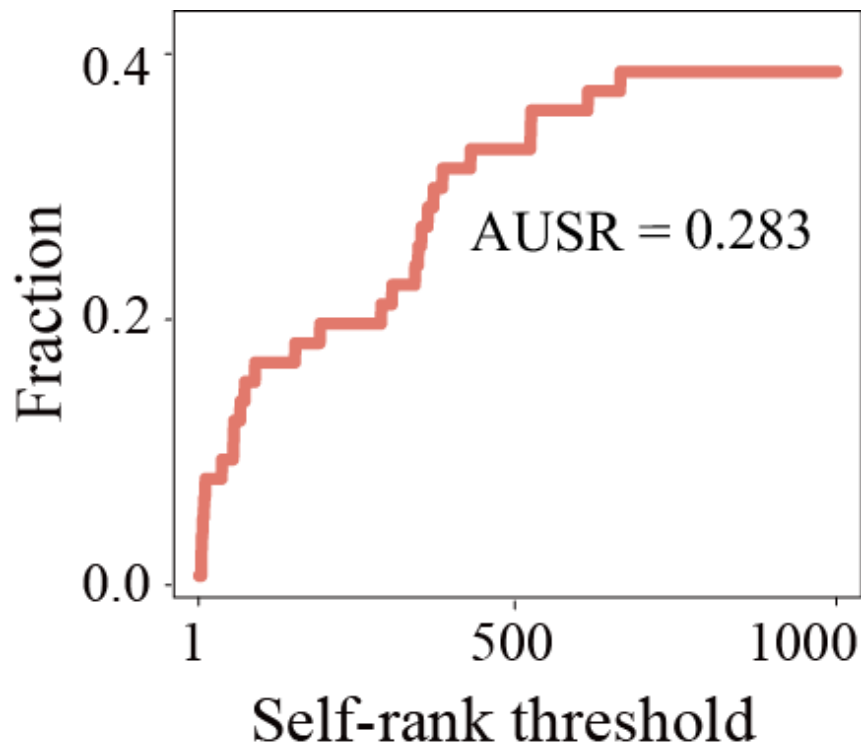
```
Zm00001d053636,Zm00001d053632,Zm00001d053630,...,Zm00001d053635
```

## 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:

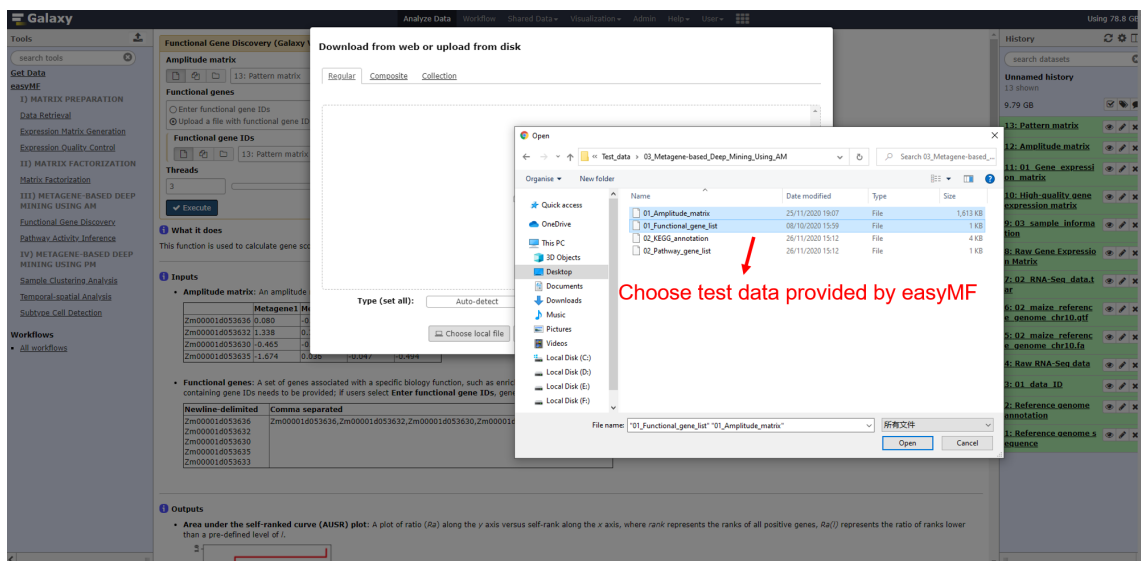
Gene ID	Score	Rank	Annotation
Zm00001d053636	1	1	Label
Zm00001d053632	0.888	3	Label
Zm00001d004839	1	1	Unlabel
...	...	...	...
Zm00001d053635	0.92	2	Unlabel

- **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  $l$ .

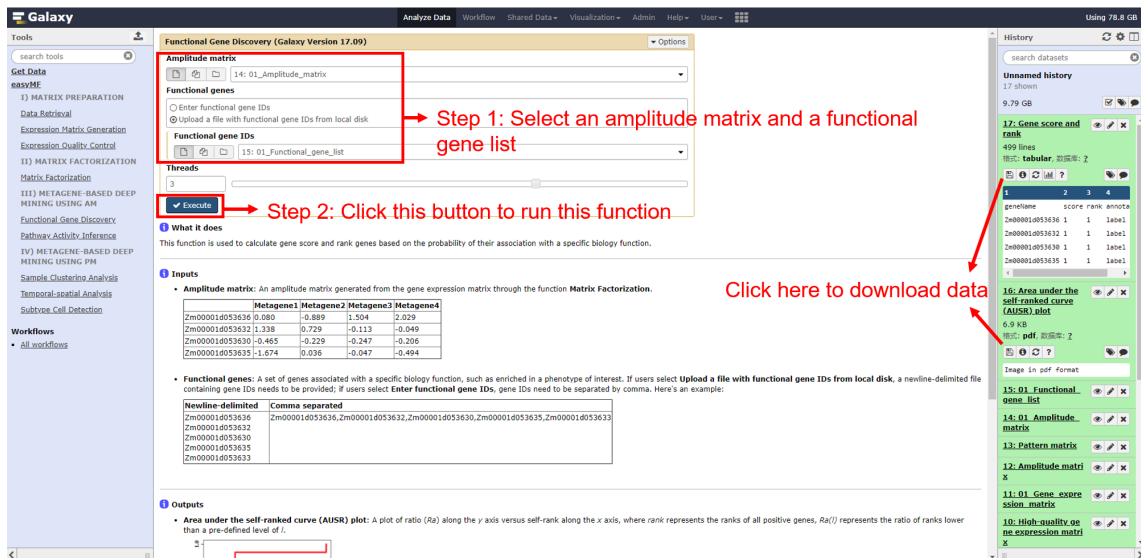


### How to use this function

- 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.

## 2. Pathway Activity Inference

Pathway activity inference can be used to examine the pathway activity for any gene set of interest.

### 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

- **Pathway annotation:** A pathway annotation file, which contains **Gene ID**, **Pathway ID**, and **Pathway name** separated by a tab character. Here is an example:

Gene ID	Pathway ID	Pathway name
Zm00001d042869	zma00010	Glycolysis / Gluconeogenesis
Zm00001d025586	zma00010	Glycolysis / Gluconeogenesis
Zm00001d039089	zma00020	Citrate cycle (TCA cycle)
Zm00001d037278	zma00030	Pentose phosphate pathway

- **Gene set:** A list of genes used to estimate pathway activity.

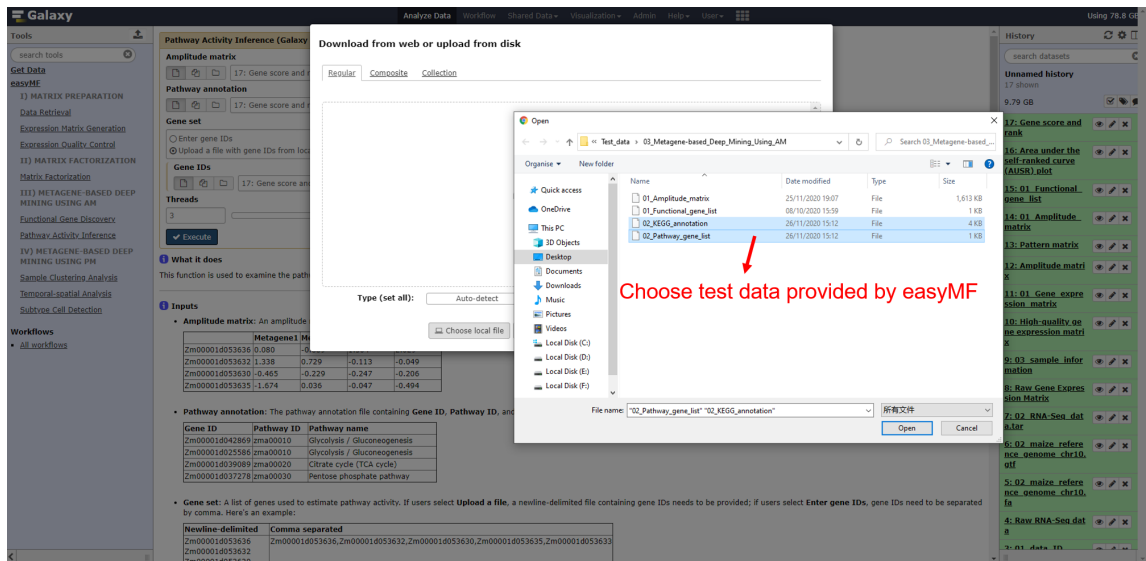
## Outputs

- **Pathway activity:** The activity of each pathway. Each column shows **Pathway**, **P-value**, **FDR**, **Term**, **Significant**, **Annotate** and **AM coefficient** in each metagene. In the result, active pathway can be obtained through a *p*-value filtration of each pathway information.

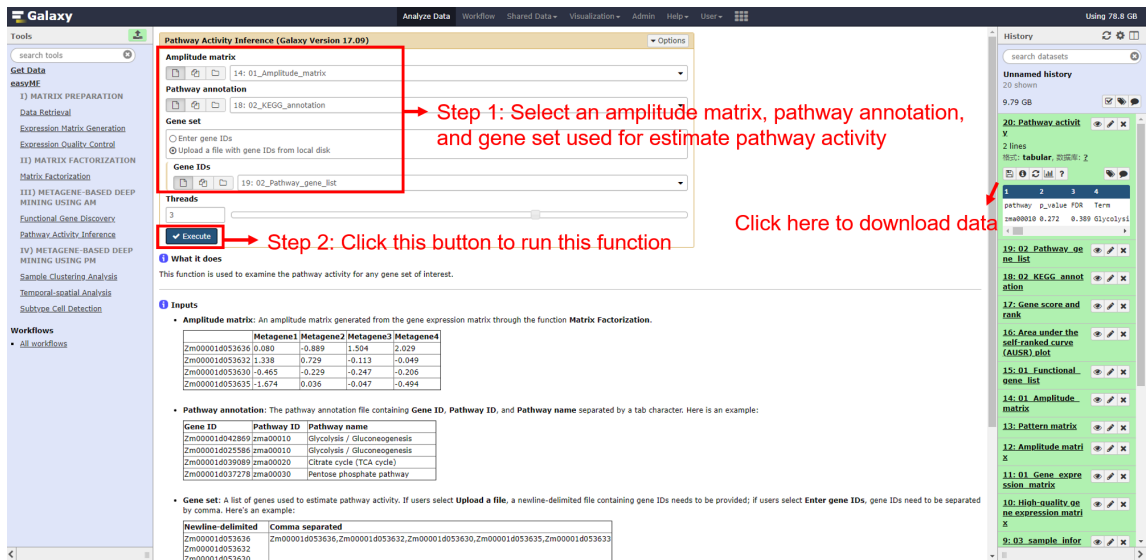
Pathway	P-value	FDR	Term	Significant	Annotate	Metagene
zma00592	0.077	0.977	alpha-Linolenic acid metabolism	3	33	0.001
zma04146	0.101	0.934	Peroxisome	2	67	0.016
zma00010	0.177	0.864	Glycolysis / Gluconeogenesis	2	102	0.031
zma00906	0.227	0.981	Carotenoid biosynthesis	2	27	0

## How to use this function

- Test data for this function are in directory `Test_data/03_Metagene-based_Deep_Mining_Using_AM` including `01_Amplitude_matrix` and `01_Functional_gene_list`.
- The following screenshots show us how to examine the pathway activity 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 ~ 1 mins for the test data.