
getENRICH: A web application for in-depth analysis of gene sets for non-model and model organisms

1. OVERVIEW

getENRICH is designed to address the unique needs of researchers working with non-model organisms and model organisms. The tool features a user-friendly interface (both GUI and CLI versions), ensuring that researchers can easily input data and interpret results. It boasts high accuracy and fast processing times, making it efficient for large datasets. Additionally, getENRICH is compatible with a wide range of non-model organisms and integrates extensively with major biological databases to ensure comprehensive and up-to-date data analysis. The development process involved collaboration with bioinformatics experts and utilised technologies such as Bash, R and R packages. The results of the enrichment analysis show a CSV file and several types of plots such as Bar Plots, BarPlot_qScore, Dot Plots, Lollipop Plots, Cnet Plots, Heatmap Plots, Upset Plots, Tree Plots, PubMed Trend Plots along with the KEGG Pathway Diagrams.

Both GUI and CLI versions are explained in detail in the following sections.

2. GUI VERSION

2.1.

3. CLI VERSION

3.1. INSTALLATION

3.1.1. PREREQUISITES:

getENRICH requires the following dependencies to be installed:

- Secure internet connection
- Bash dependencies:
 - wget
 - jq
- R dependencies:
 - R 4.4.1
 - Required libraries:
 - jsonlite 1.8.8
 - dplyr 1.1.4
 - tidyverse 2.0.0
 - clusterProfiler 4.12.0
 - pheatmap 1.0.12
 - visNetwork 2.1.2
 - enrichplot 1.24.0
 - ggplot2 3.5.1
 - UpSetR 1.4.0
 - pathview 1.44.0
 - plotly 4.10.4

- cowplot 1.1.3

3.1.2. INSTALLATION STEPS:

Clone the GitHub repository using:

git clone <https://github.com/jnarayan81/getENRICH.git>

3.2. PREPARATION OF INPUT FILES

The protein FASTA files can be downloaded from the NCBI website, other sources, or provided by the users themselves if they have newly sequenced protein data.

To perform enrichment analysis using getENRICH, users are required to provide three input files:

1. **Background Genes:** A file containing the background gene set.
2. **Genes of Interest:** A file containing the genes of interest.
3. **Annotated Genes:** A file with the annotated gene information in a specific format.

The IDs of the background gene set and genes of interest should be same as protein sequence IDs in FASTA file.



```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/input$ ls
1background_genes.sb 2genes_of_interest.sb 3kegg_annotationT0genes.sb
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/input$ head 1background_genes.sb
gene
UJR06542
UJR06549
UJR06551
UJR06566
UJR06567
UJR06589
UJR06619
UJR06636
UJR06638
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/input$ head 2genes_of_interest.sb
gene
UJR06642
UJR06644
UJR06811
UJR06812
UJR06813
UJR06829
UJR06830
UJR06837
UJR06838
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/input$ head 3kegg_annotationT0genes.sb
term    gene
K02927  UJR06542
K13075  UJR06549
K00503  UJR06551
K15266  UJR06551
K05970  UJR06566
K05970  UJR06567
K12440  UJR06589
K12441  UJR06589
K01275  UJR06619

```

Format of input file containing background genes

Format of input file containing genes of interest

Format of input file containing annotated genes

Figure 1 Screenshot of the format of three input files

3.2.1. GENERATING ANNOTATED FILES

To create the annotated files, users must follow these steps:

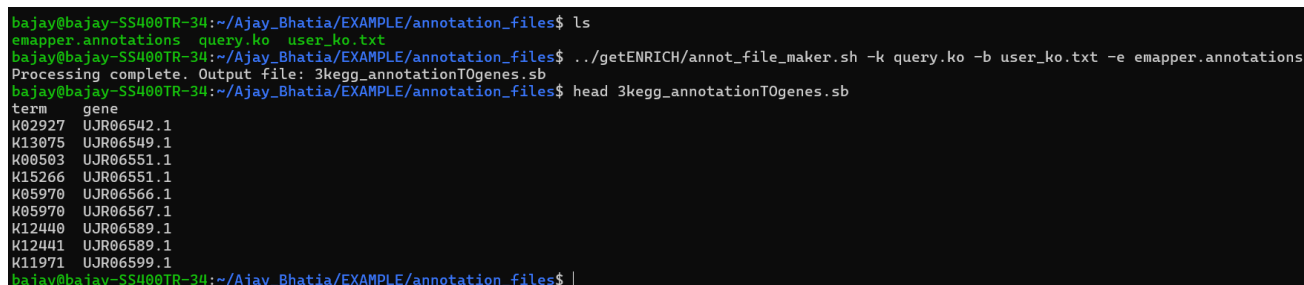
1. **Use the Annotation Tools:** The protein FASTA files can be annotated using the following annotation tools or any other annotation tools:
 - BlastKOALA (<https://www.kegg.jp/blastkoala/>): Provides KEGG orthology assignments.

- eggNOG-mapper (<http://eggno-mapper.embl.de/>): Offers functional annotations and orthology assignments.
- KAAS-KEGG Automatic Annotation Server (<https://www.genome.jp/kegg/kaas/>): Provides functional annotation of genes by KEGG Orthology.

Users can choose to use one, two or any number of annotation tools to annotate their files.

2. **Format the Annotated Files:** After annotation, the files need to be converted to a specific format required by getENRICH (refer to Figure 1 for the format). The script `annot_file_maker.sh` (in the getENRICH folder on the GitHub repository) can be used to convert the annotated files to the desired format. The tool supports three flags to indicate which annotation tool was used:

- `-b` for BlastKOALA
- `-k` for KAAS-KEGG Automatic Annotation Server
- `-e` for eggNOG-mapper



```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/annotation_files$ ls
emapper.annotations query.ko user.ko.txt
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/annotation_files$ ../getENRICH/annot_file_maker.sh -k query.ko -b user.ko.txt -e emapper.annotations
Processing complete. Output file: 3kegg_annotationT0genes.sb
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/annotation_files$ head 3kegg_annotationT0genes.sb
term    gene
K02927  UJR06542.1
K13075  UJR06549.1
K00503  UJR06551.1
K15266  UJR06551.1
K05970  UJR06566.1
K05970  UJR06567.1
K12440  UJR06589.1
K12441  UJR06589.1
K11971  UJR06599.1
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/annotation_files$ |

```

Figure 2 Screenshot of command-line usage for the script `annot_file_maker.sh`

To use the script `annot_file_maker.sh`, users need to use the command to format and merge the annotated files from all three annotation tools is as follows:

```
../getENRICH/annot_file_maker.sh -k query.ko -b user_ko.txt -e
emapper.annotations
```

Here:

- `-k query.ko`: Specifies the file `query.ko` is annotated with KAAS-KEGG Automatic Annotation Server.
- `-b user_ko.txt`: Specifies the file `user_ko.txt` is annotated with BlastKOALA.
- `-e emapper.annotations`: Specifies the file `emapper.annotations` is annotated with eggNOG-mapper.

You can use one, two, or all three flags based on the tools you have used for annotation. For instance:

- If only BlastKOALA is used, the command would be:

```
../getENRICH/annot_file_maker.sh -b user_ko.txt
```

- If BlastKOALA and eggNOG-mapper is used, the command would be:

```
../getENRICH/annot_file_maker.sh -b user_ko.txt -e emapper.annotations
```

This process will not only bring the annotated file to the desired format but will also merge the results of annotations obtained from all the specified tools.

By following these steps, users can ensure that their annotated files are correctly formatted and ready for use with the getENRICH tool for gene enrichment analysis.

3.3. HOW TO USE THE TOOL?

```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE$ ls
annotation_files  getENRICH  input
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE$ cd getENRICH/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ ls
annot_file_maker.sh  config.json  getENRICH  ScriptBase
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ nano config.json
  
```

Figure 3 Screenshot of the CLI version guide. The red marked boxes respectively indicate Step 1, 2 and 3.

Step 1: Open the tool. Use the command `cd getENRICH` to open the tool.

Step 2: Use the command `ls` to list all the files and programs in the tool.

Step 3: Then open the configuration file with the command `nano config.json`.

The `config.json` file will open showing the location of the three input files (consisting of background genes, genes of interest, and the KEGG annotation genes) the names of which the users can edit according to their convenience. They can check the location of these configuration files and then save these files. It should be noted that output files and the scripts of the tool are also editable. So, the users should keep in mind not to edit the output files or the scripts when dealing with the input files.

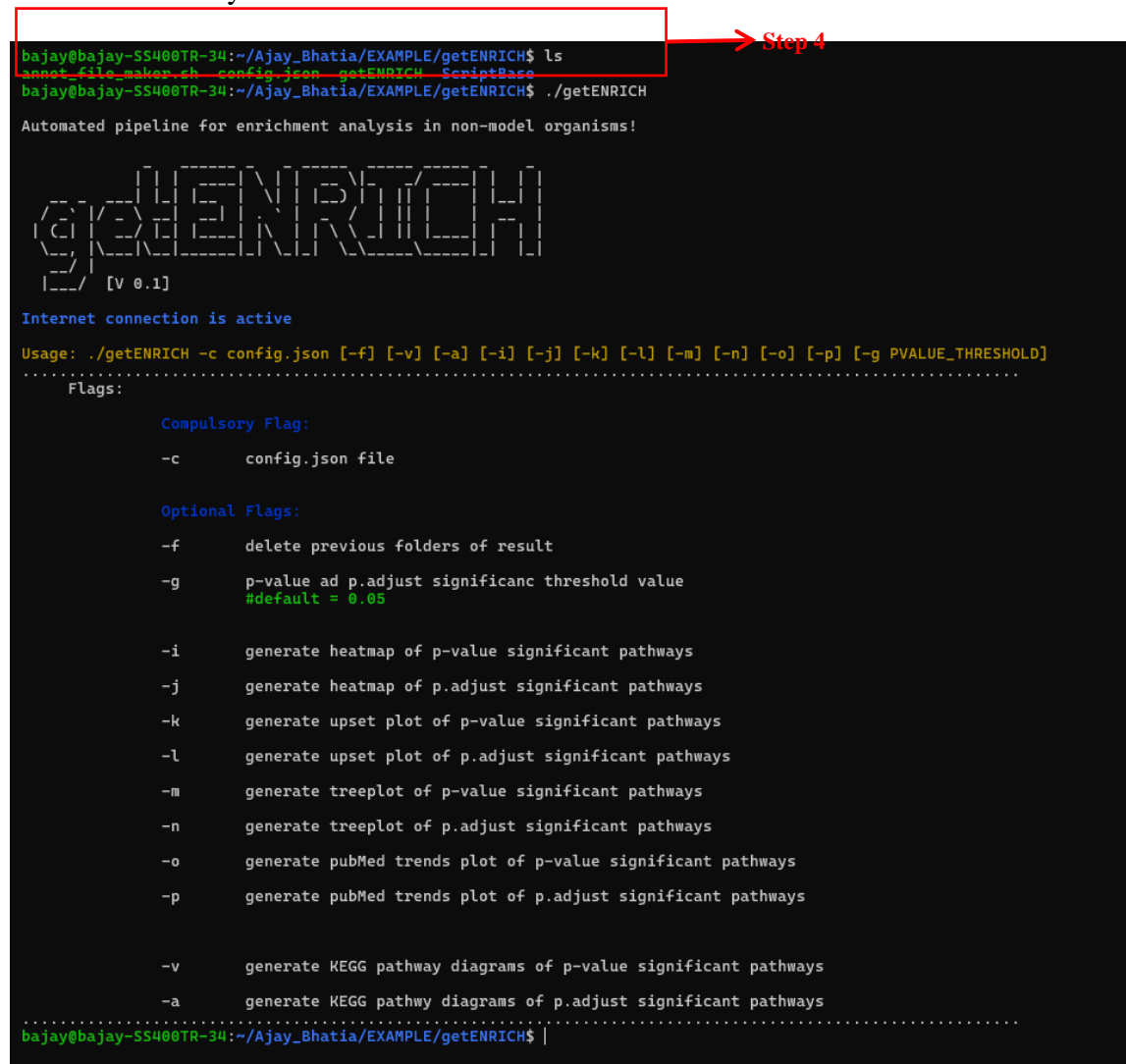
```

GNU nano 6.2 config.json
{
  "input_files": {
    "background_genes_sb_1": "/home/bajay/Ajay_Bhatia/EXAMPLE/input/1background_genes.sb",
    "kegg_annotationT0genes_sb_3": "/home/bajay/Ajay_Bhatia/EXAMPLE/input/3kegg_annotationT0genes.sb",
    "genes_of_interest_sb_2": "/home/bajay/Ajay_Bhatia/EXAMPLE/input/2genes_of_interest.sb"
  },
  "output_files": {
    "outdir": "/home/bajay/Ajay_Bhatia/EXAMPLE/out",
    "graph": "graph",
    "pathway": "pathway",
    "enrichment_KEGG_results_csv": "enrichment_KEGG_results.csv"
  },
  "scripts": {
    "clusterProfiler_script": "ScriptBase/clusterProfiler_script.R",
    "pathview_graph_based_on_p_adjust": "ScriptBase/pathview_graph_based_on_p_adjust.R",
    "pathview_graph_based_on_pvalue": "ScriptBase/pathview_graph_based_on_pvalue.R",
    "plot_enrichment_graphs": "ScriptBase/plot_enrichment_graphs.R",
    "pubMed_trends_plots_padjust": "ScriptBase/pubMed_trends_plots_padjust.R",
    "pubMed_trends_plots_pvalue": "ScriptBase/pubMed_trends_plots_pvalue.R",
    "treeplot_padjust": "ScriptBase/treeplot_padjust.R",
    "treeplot_pvalue": "ScriptBase/treeplot_pvalue.R",
    "upset_plot_pvalue": "ScriptBase/upset_plot_pvalue.R",
    "upset_plot_padjust": "ScriptBase/upset_plot_padjust.R",
    "heatmap_plot_padjust": "ScriptBase/heatmap_plot_padjust.R",
    "heatmap_plot_pvalue": "ScriptBase/heatmap_plot_pvalue.R"
  }
}
  
```

Figure 4 Screenshot of the output of Step 3, i.e., the `config.json` file being opened and showing the location of the input files, the output files, and the scripts of the tool

Step 4: Use the command `./getENRICH` to access the usage of the tool.

This is the help command. Alternatively, you can use the command `bash getENRICH` to perform the same function. The above output is generated which is an automated pipeline for enrichment analysis in non-model organisms. It lists all the flags and their functions, which the user may need for performing their desired analysis.



```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ ls
annot_file_maker.sh config.json getENRICH ScriptBase
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ ./getENRICH

Automated pipeline for enrichment analysis in non-model organisms!

  getENRICH
  [V 0.1]

Internet connection is active

Usage: ./getENRICH -c config.json [-f] [-v] [-a] [-i] [-j] [-k] [-l] [-m] [-n] [-o] [-p] [-g PVALUE_THRESHOLD]
Flags:
.....
Compulsory Flag:
-c      config.json file

Optional Flags:
-f      delete previous folders of result
-g      p-value ad p.adjust significanc threshold value
        #default = 0.05
-i      generate heatmap of p-value significant pathways
-j      generate heatmap of p.adjust significant pathways
-k      generate upset plot of p-value significant pathways
-l      generate upset plot of p.adjust significant pathways
-m      generate treepplot of p-value significant pathways
-n      generate treepplot of p.adjust significant pathways
-o      generate pubMed trends plot of p-value significant pathways
-p      generate pubMed trends plot of p.adjust significant pathways

-v      generate KEGG pathway diagrams of p-value significant pathways
-a      generate KEGG pathwy diagrams of p.adjust significant pathways
.....
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ |

```

Figure 5 Screenshot of Step 4. The red marked box indicating Step 4 and the script below the red box is the output of Step 4 which lists out all the flags used by the tool along with their functions.

3.3.1. DETAILED EXPLANATION OF FLAGS FOR GETENRICH

The getENRICH tool uses various flags to customise its operation. Below is a detailed explanation of each flag:

3.3.1.1. COMPULSORY FLAG:

1. -c (config.json file)

This flag specifies the configuration file that contains the necessary parameters and settings for running the tool. The `config.json` file should be formatted correctly and include all required fields. It is the compulsory flag and the minimum flag the user requires to run the enrichment analysis.

Usage Example:

```
./getENRICH -c config.json
```

3.3.1.2. OPTIONAL FLAGS:

1. **-f (delete previous folders of result)**

This flag allows the user to delete any previous result folders before running a new analysis. This can be useful for ensuring that old results do not interfere with new ones. If this flag is not used, then the old folder will be over-written with the contents of the new folder.

Usage Example:

```
./getENRICH -c config.json -f
```

2. **-g (p-value and p.adjust significance threshold value)**

This flag sets the significance threshold for both p-value and adjusted p-value. The default value is 0.05, but the user can specify a different threshold if needed. If there are no P-value or P-adjust values below the set threshold in the resultant Excel file generated, blank plots for P-adjust and P-value will be generated.

Usage Example:

```
./getENRICH -c config.json -g 0.01
```

In this example, the threshold is set to 0.01, so all the P-values and adjusted P-values which are below 0.01 will be plotted in the resultant graphs and plots and if there are no P-value or adjusted P-values below the set threshold in the resultant Excel file generated, blank plots for P-adjust and P-value will be generated.

3. **-i (generate heatmap of p-value significant pathways)**

This flag generates a heatmap for pathways that are significant based on the p-value. This visualisation helps in identifying patterns and clusters of significant pathways.

Usage Example:

```
./getENRICH -c config.json -i
```

4. **-j (generate heatmap of p.adjust significant pathways)**

Similar to -i, this flag generates a heatmap but for pathways significant based on the adjusted p-value.

Usage Example:

```
./getENRICH -c config.json -j
```

5. **-k (generate upset plot of p-value significant pathways)**

This flag generates an upset plot for pathways significant based on the p-value. Upset plots are useful for visualising the intersections of sets.

Usage Example:

```
./getENRICH -c config.json -k
```

6. -l (generate upset plot of p.adjust significant pathways)

Similar to -k, this flag generates an upset plot but for pathways significant based on the adjusted p-value.

Usage Example:

```
./getENRICH -c config.json -l
```

7. -m (generate treeplot of p-value significant pathways)

This flag generates a tree plot for pathways significant based on the p-value. Tree plots help in visualising hierarchical relationships.

Usage Example:

```
./getENRICH -c config.json -m
```

8. -n (generate treeplot of p.adjust significant pathways)

Similar to -m, this flag generates a tree plot but for pathways significant based on the adjusted p-value.

Usage Example:

```
./getENRICH -c config.json -n
```

9. -o (generate pubMed trends plot of p-value significant pathways)

This flag generates a PubMed trends plot for pathways significant based on the p-value. This plot shows the publication trends over time for the significant pathways.

Usage Example:

```
./getENRICH -c config.json -o
```

10. -p (generate pubMed trends plot of p.adjust significant pathways)

Similar to -o, this flag generates a PubMed trends plot but for pathways significant based on the adjusted p-value.

Usage Example:

```
./getENRICH -c config.json -p
```

11. -v (generate KEGG pathway diagrams of p-value significant pathways)

This flag generates KEGG pathway diagrams for pathways significant based on the p-value. KEGG pathway diagrams are useful for understanding the biological pathways involved.

Usage Example:

```
./getENRICH -c config.json -v
```

12. -a (generate KEGG pathway diagrams of p.adjust significant pathways)

Similar to -v, this flag generates KEGG pathway diagrams but for pathways significant based on the adjusted p-value.

Usage Example:

```
./getENRICH -c config.json -a
```

3.4. USAGE

3.4.1. EXAMPLE 1: COMMAND WITH ONLY THE COMPULSORY FLAG

- **Running the tool**

The user can run the command `./getENRICH -c config.json` as the basic minimum command to perform the enrichment analysis. This will generate some plots by default. This can be explained as follows.

```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ ls
annot_file_maker.sh  config.json  getENRICH  ScriptBase
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ ./getENRICH -c config.json

Automated pipeline for enrichment analysis in non-model organisms!

getENRICH
[ v 0.1 ]

Internet connection is active

--:LOCATIONS and FLAGS:--
CONFIG FILE           : config.json
FORCE FLAG            : 0
p.adjust PUBMED TREND PLOT flag : 0
p-value PUBMED TREND PLOT flag : 0
p-value KEGG PATHWAY   : 0
p.adjust KEGG PATHWAY  : 0
p-value/p.adjust THRESHOLD : 0.05
p-value HEATMAP        : 0
p.adjust HEATMAP       : 0
p-value UPSETPLOT      : 0
p.adjust UPSETPLOT     : 0
p.adjust TREEPLOT      : 0
p-value TREEPLOT       : 0

Do you want to proceed with running the pipeline? (y/n): y
Proceeding with the pipeline...

[2024-07-12 10:03:48] Calculating Enrichment of KEGG Pathway
[2024-07-12 10:04:16] Enrichment Analysis Completed
[2024-07-12 10:04:16] Generating General Plots of Enrichment Result
[2024-07-12 10:04:26] General Plots of Enrichment Result Generated

Enrichment analysis completed successfully
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$

```

Example Command with only the Compulsory Flag

Figure 6 Screenshot of the example command with only the compulsory flag. The script below the command shows the output of the command. 1- Command Initialization and Configuration, 2- Internet Connectivity Checkup, 3- Initial Setup and Cleanup, 4- Permission to run the analysis, 5- Generating Plots, 6- Completion

The output of the command with only the compulsory flag can be described as follows (refer to Figure 4).

1. Command Initialization and Configuration:

The tool initializes and confirms the start of the enrichment analysis pipeline.

2. Internet Connectivity Checkup:

The tool checks if the user is connected to the internet or not. If the user is connected to the internet then the tool proceeds to execute the command by printing “Internet connection is active”, otherwise the tool wouldn’t execute the command.

3. Initial Setup and Cleanup:

The tool lists the configuration settings and flags used. The force flag (`-f`) is not confirmed, indicating that previous output files are not cleaned up. For a flag used the output will print 1 otherwise it will print 0 and for the p-value/p-adjust threshold will portray the set threshold value. Here, only the compulsory flag is being used.

4. Permission to run the analysis:

The tool takes confirmation from the user if they want to continue running the pipeline to continue the enrichment analysis. If the user writes “y”, “Y”, “yes”, “Yes” or “YES” then the tool will proceed to run the enrichment analysis and otherwise if the user denies permission and writes “n”, “N”, “no”, “No”, or “NO” then the tool will not run the analysis.

5. Generating Plots:

- **Enrichment Analysis:**

The tool performs the enrichment analysis of KEGG pathways based on the provided genomic data and parameters. This step involves identifying significant pathways and calculating their enrichment scores. This also shows the exact time when this step is performed.

- **Generating General Plots:**

General plots are generated to provide an overview of the enrichment results. These might include bar plots or other summary visualisations. This also shows the exact time when this step is performed.

6. Completion:

The tool completes the enrichment analysis and confirms successful execution. All specified plots are generated based on the analysis results.

- **Summary of Plots Generated (by this command)**

This command will generate 10 plots in total, comprising five types of plots: Bar Plot, BarPlot_qScore, Dot Plot, Lollipop Plot, and Cnet Plot. All the plots are generated for both P-value and adjusted P-value. The default significance threshold for both P-value and P-adjust is set to 0.05. Therefore, even when generating the default plots, the tool will use a P-value and P-adjust threshold of 0.05. If there are no P-value or P-adjust values below 0.05 in the resultant CSV file generated, blank plots for P-adjust and P-value will be generated.

- **Bar Plots:** General visualizations of significant pathways.
- **BarPlot_qScore:** Visualizations of pathways based on adjusted p-values.
- **Dot Plots:** Dots representing pathways with significance and enrichment scores.
- **Lollipop Plots:** Hybrid of bar and dot plots for clear visualization.
- **Cnet Plots:** Network visualizations of gene-pathway relationships.

These plots provide a comprehensive view of the enrichment analysis results, aiding in the interpretation and understanding of the biological significance of the findings.

■ How to view the enrichment analysis results?

To view the results of the enrichment analysis, follow the commands in Figure 5. The user has to go back to the previous directory using the command `cd ..` and then change the directory using the command `cd` and go to the directory named `out` to access the output files. The user can list all the files present in the current directory by using the command `ls`. The result shows a CSV file named “enrichment_KEGG_results.csv” and a folder named “graph” where all the plots are present. The user can again change the directory using the command `cd graph` and list all the files present by using the command `ls` to access all the plots from the enrichment analysis.

```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ ls
annot_file_maker.sh  config.json  getENRICH  ScriptBase
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/getENRICH$ cd ..
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE$ ls
annotation_files  getENRICH  input  out
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE$ cd out/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ ls
enrichment_KEGG_results.csv  graph
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ cd graph/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/graph$ ls
p_adjust_bar_plot.html          padjust_cnetplot_interactive.html  padjust_lollipop_plot.png          p_value_bar_plot_qscore.png          p_value_lollipop_plot.html
p_adjust_bar_plot.png           padjust_dot_plot.html              p_value_bar_plot.png              pvalue_cnetplot_interactive.html      p_value_lollipop_plot.png
p_adjust_bar_plot_qscore.html   padjust_dot_plot.png              p_value_bar_plot_qscore.html      p_value_dot_plot.html
p_adjust_bar_plot_qscore.png    padjust_lollipop_plot.html        p_value_bar_plot_qscore.html      p_value_dot_plot.png
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/graph$ |

```

Figure 7 Screenshots of the steps to view the results of the enrichment analysis for example 1

The user can also view the first 10 contents of the CSV file by using the command `head enrichment_KEGG_results.csv` when in the `out` directory (refer to Figure 6)

```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ ls
enrichment_KEGG_results.csv  graph
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ head enrichment_KEGG_results.csv
**"category","subcategory","ID","Description","GeneRatio","BgRatio","pvalue","p.adjust","qvalue","geneID","Count"
"ko08599","Metabolism","Lipid metabolism","ko08599","Arachidonic acid metabolism","5/47","25/4019",0.00841550657595e-06,0.008816235212683899,0.008735272610616063,"ko1832","K18592","K01254","K17726","K17728",5
"ko082818","Environmental Information Processing","Membrane transport","ko082818","ABC transporters","5/47","41/4019",0.62451225121204e-05,0.00580474637063026,0.00458832415977827,"K11804","K05658","K05659","K18104","K0
2836",5
"ko04112","Cellular Processes","Transport and catabolism","ko04112","Lysosome","16/47","61/4019",0.0003080800710275535,0.0107050915660807,0.00044325050053737,"ko1365","K01363","K01371","K01375","K09599","K09600",6
"ko080408","Metabolism","Metabolism of other amino acids","ko080408","Glutathione metabolism","4/47","32/4019",0.00044663266669080,0.0120884003341742,0.010821061411083,"K10592","K08601","K08797","K11284",4
"ko040976","Organismal Systems","Digestive system","ko040976","Bile secretion","4/47","39/4019",0.00100094172983776,0.0208195879639855,0.0187540871335497,"K14349","K05658","K05659","K14158",4
"ko04210","Cellular Processes","Cell growth and death","ko04210","Apoptosis","4/47","57/4019",0.00412213248864161,0.0714502940316545,0.0643631212795269,"K01365","K01368","K01371","K01375",4
"ko08561","Metabolism","Lipid metabolism","ko08561","Glycerolipid metabolism","3/47","39/4019",0.0102041063920919,0.151603866396794,0.136566235924238,"K07407","K14073","K14076",3
"ko04212","Organismal Systems","Aging","ko04212","Longevity regulating pathway - worm","4/47","41/4019",0.0117102623407104,0.152233410429235,0.137133335305607,"K08341","K14035","K11284",3
"ko04139","Cellular Processes","Transport and catabolism","ko04139","Mitophagy - yeast","2/47","17/4019",0.016281515322182,0.18131113845354,0.1632683524661,"K17065","K08341",2
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ |

```

Figure 8 Screenshot of the steps to view the first 10 contents of the CSV file

Category	Subcategory	ID	Description	GeneRatio	BiRatio	pvalue	p.adjust	qvalue	geneID	Count
ko00590	Metabolism	ko00590	Ascorbate acid metabolism	5/47	25/4019	7.84841550657595E-06	0.000816235212663899	0.000735272610616083	K01832K11850K01254K17726K17728	5
ko02010	Environmental Information Processing	ko02010	Membrane transport	5/47	41/4019	9.62451225121204E-05	0.00500474637063026	0.0049032615977827	K11004K05658K05659K18104K02036	5
ko04142	Cellular Processes	ko04142	Transport and catabolism	6/47	81/4019	0.000308800718279635	0.010705991568887	0.009643295053737	K01365K01368K01371K01375K09599K09600	6
ko00490	Metabolism	ko00490	Metabolism of other amino acids	4/47	32/4019	0.00046483266699006	0.0120844924142	0.010862196141083	K18950K00681K00707K11204	4
ko04976	Organismal Systems	ko04976	Digestive system	4/47	39/4019	0.0010094172903776	0.0208195879639855	0.018758471335497	K14349K05658K05659K14158	4
ko04210	Cellular Processes	ko04210	Cell growth and death	4/47	57/4019	0.00412213248644161	0.0714502964316545	0.064363121795269	K01365K01368K01371K01375	4
ko00561	Metabolism	ko00561	Lipid metabolism	3/47	39/4019	0.0102041063920919	0.151602866396794	0.13666623504238	K07407K14073K14078	3
ko04212	Organismal Systems	ko04212	Aging	3/47	41/4019	0.0117103623407104	0.192333410429295	0.13713333935687	K08341K14035K11204	3
ko04139	Cellular Processes	ko04139	Transport and catabolism	2/47	17/4019	0.016281515322182	0.1813113845354	0.16332683524661	K17065K08341	2
ko04612	Organismal Systems	ko04612	Immune system	2/47	18/4019	0.0181814901179841	0.1813113845354	0.16332683524661	K01365K01368	2
ko05223	Human Diseases	ko05223	Immune disease	2/47	19/4019	0.0201700071304459	0.1813113845354	0.16332683524661	K01365K01371	2
ko04621	Organismal Systems	ko04621	Immune system	3/47	52/4019	0.02204472377036	0.1813113845354	0.16332683524661	K17065K08341K09849	3
ko00360	Metabolism	ko00360	Amino acid metabolism	2/47	21/4019	0.024072686379766	0.1813113845354	0.16332683524661	K1075K00500	2
ko00700	Metabolism	ko00700	Metabolism of cofactors and vitamins	2/47	21/4019	0.024072686379766	0.1813113845354	0.16332683524661	K00502K00500	2
ko00830	Metabolism	ko00830	Metabolism of cofactors and vitamins	2/47	30/4019	0.0477168034369562	0.32806317049555	0.295522491640729	K07424K00157	2
ko00380	Metabolism	ko00380	Amino acid metabolism	2/47	36/4019	0.055848195653078	0.40122477863953	0.361427179138848	K00502K00157	2
ko04668	Environmental Information Processing	ko04668	Signal transduction	2/47	36/4019	0.055848195653078	0.40122477863953	0.361427179138848	K17065K09849	2
ko04214	Cellular Processes	ko04214	Cell growth and death	2/47	41/4019	0.0823122657150224	0.46455380855036	0.418474584719618	K03262K17065	2
ko00270	Metabolism	ko00270	Amino acid metabolism	2/47	42/4019	0.086824326239108	0.46455380855036	0.418474584719618	K00707K11204	2
ko04217	Cellular Processes	ko04217	Cell growth and death	2/47	43/4019	0.089372708951993	0.46455380855036	0.418474584719618	K17065K09849	2
ko00591	Metabolism	ko00591	Lipid metabolism	1/47	10/4019	0.111096317731469	0.502348567133597	0.452502407039325	K07424	1
ko00204	Cellular Processes	ko00204	Quorum sensing	1/47	10/4019	0.111096317731469	0.502348567133597	0.452502407039325	K00218	1
ko04977	Organismal Systems	ko04977	Digestive system	1/47	10/4019	0.111096317731469	0.502348567133597	0.452502407039325	K14073	1
ko04145	Cellular Processes	ko04145	Transport and catabolism	2/47	53/4019	0.126832604805069	0.5496079541553	0.495002185423296	K01365K01368	2
ko02020	Environmental Information Processing	ko02020	Signal transduction	1/47	14/4019	0.15206873277578	0.5681102771196	0.5117952626798	K01179	1
ko04973	Organismal Systems	ko04973	Digestive system	1/47	14/4019	0.15206873277578	0.5681102771196	0.5117952626798	K014158	1
ko00340	Metabolism	ko00340	Amino acid metabolism	1/47	15/4019	0.16201948672223	0.5681102771196	0.5117952626798	K01745	1
ko00962	Metabolism	ko00962	Xenobiotics biodegradation and metabolism	1/47	16/4019	0.17185921319636	0.5681102771196	0.5117952626798	K00157	1
ko04975	Organismal Systems	ko04975	Digestive system	1/47	16/4019	0.17185921319636	0.5681102771196	0.5117952626798	K14073	1
ko00705	Metabolism	ko00705	Metabolism of cofactors and vitamins	1/47	17/4019	0.181579321693849	0.5681102771196	0.5117952626798	K00161	1
ko00920	Metabolism	ko00920	Energy metabolism	1/47	17/4019	0.181579321693849	0.5681102771196	0.5117952626798	K00988	1
ko04979	Organismal Systems	ko04979	Digestive system	1/47	17/4019	0.181579321693849	0.5681102771196	0.5117952626798	K14349	1
ko01120	Metabolism	ko01120	Microbial metabolism in diverse environments	1/47	194/4019	0.18965195962144	0.5681102771196	0.5117952626798	K00988K00181K00126K00157	1
ko04216	Cellular Processes	ko04216	Cell growth and death	1/47	18/4019	0.19119056845371	0.5681102771196	0.5117952626798	K11204	1
ko05204	Human Diseases	ko05204	Cancer: overview	1/47	18/4019	0.19119056845371	0.5681102771196	0.5117952626798	K07424	1
ko05133	Human Diseases	ko05133	Infectious disease: bacterial	1/47	19/4019	0.2006208997962	0.574966621186154	0.517939519046464	K11004	1
ko00140	Metabolism	ko00140	Lipid metabolism	1/47	20/4019	0.210383957741095	0.574966621186154	0.517939519046464	K07424	1
ko04630	Environmental Information Processing	ko04630	Signal transduction	1/47	20/4019	0.210383957741095	0.574966621186154	0.517939519046464	K00157	1
ko04136	Cellular Processes	ko04136	Transport and catabolism	1/47	21/4019	0.219367792196251	0.584980779190002	0.52699636991812	K08341	1
ko00230	Metabolism	ko00230	Nucleotide metabolism	2/47	77/4019	0.227061011736344	0.585259178309422	0.527207154548995	K00988K00106	2

Figure 9 Screenshot of the resultant CSV file

3.4.2. EXAMPLE 2: COMMAND WITH MULTIPLE FLAGS

■ Running the tool

When the user runs the command `./getENRICH -c config.json -f -i -j -k -l -m -n -o -p -v -a -g 0.05`, they are instructing the `getENRICH` tool to perform enrichment analysis with several specific settings and generate multiple types of plots. This command uses all the available flags and this can be explained as follows.

```

bajay@bajay-S5400TR-34:~/Ajay.Bhatia/EXAMPLE/getENRICH$ ls
annot_file_maker.sh config.json getENRICH ScriptBase
bajay@bajay-S5400TR-34:~/Ajay.Bhatia/EXAMPLE/getENRICH$ ./getENRICH -c config.json -f -i -j -k -l -m -n -o -p -v -a -g 0.05

Automated pipeline for enrichment analysis in non-model organisms!

getENRICH
[1] [v 0.1]

Internet connection is active

---LOCATIONS and FLAGS---
CONFIG FILE           : config.json
FORCE FLAG            : 1
p.adjust PUBMED TREND PLOT flag : 1
p-value PUBMED TREND PLOT flag : 1
p-value KEGG PATHWAY    : 1
p.adjust KEGG PATHWAY   : 1
p-value/p.adjust THRESHOLD : 0.05
p-value HEATMAP         : 1
p.adjust HEATMAP        : 1
p-value UPSETPLOT       : 1
p.adjust UPSETPLOT      : 1
p-value TREEPLOT        : 1
p.adjust TREEPLOT       : 1

Do you want to proceed with running the pipeline? (y/n): y
Proceeding with the pipeline

Force flag is set. Cleaning up previous output files and directories...

[2024-07-12 10:08:05] Calculating Enrichment of KEGG Pathway
[2024-07-12 10:08:23] Enrichment Analysis Completed
[2024-07-12 10:08:23] Generating General Plots of Enrichment Result
[2024-07-12 10:08:31] General Plots of Enrichment Result Generated
[2024-07-12 10:08:31] Generating PubMed Trend Plots of p.adjust Significant Enriched Pathways
[2024-07-12 10:09:56] PubMed Trend Plots of p.adjust Significant Enriched Pathways Generated
[2024-07-12 10:10:26] Generating PubMed Trend Plots of p-value Significant Enriched Pathways
[2024-07-12 10:13:50] PubMed Trend Plots of p-value Significant Enriched Pathways Generated
[2024-07-12 10:13:50] Generating p.adjust Significant Pathway Diagrams
[2024-07-12 10:14:44] p.adjust Significant Pathway Diagrams Generated
[2024-07-12 10:14:44] Generating p-value Significant Pathway Diagrams
[2024-07-12 10:16:24] p-value Significant Pathway Diagrams Generated
[2024-07-12 10:16:24] Generating Heatmap Plot of p-value Significant Enriched Pathways
[2024-07-12 10:16:28] Heatmap Plot of p-value Significant Enriched Pathways Generated
[2024-07-12 10:16:28] Generating Heatmap Plot of p.adjust Significant Enriched Pathways
[2024-07-12 10:16:33] Heatmap Plot of p.adjust Significant Enriched Pathways Generated
[2024-07-12 10:16:33] Generating Upset Plot of p-value Significant Enriched Pathways
[2024-07-12 10:16:39] Upset Plot of p-value Significant Enriched Pathways Generated
[2024-07-12 10:16:39] Generating Upset Plot of p.adjust Significant Enriched Pathways
[2024-07-12 10:16:44] Upset Plot of p.adjust Significant Enriched Pathways Generated
[2024-07-12 10:16:49] Generating Treemap Plot of p-value Significant Enriched Pathways
[2024-07-12 10:16:49] Treemap Plot of p-value Significant Enriched Pathways Generated
[2024-07-12 10:16:54] Treemap Plot of p-value Significant Enriched Pathways Generated

Enrichment analysis completed successfully
bajay@bajay-S5400TR-34:~/Ajay.Bhatia/EXAMPLE/getENRICH$

```

Example Command with Multiple Flags

1

2

3

4

5

6

Figure 10 Screenshot of the example command with multiple flags. The script below the command shows the output of the command. 1- Command Initialization and Configuration, 2- Internet Connectivity Checkup, 3- Initial Setup and Cleanup, 4- Permission to run the analysis, 5- Generating Plots, 6- Completion

1. Command Initialization and Configuration:

The tool initialises and confirms the start of the enrichment analysis pipeline.

2. Internet Connectivity Checkup:

The tool checks if the user is connected to the internet or not. If the user is connected to the internet then the tool proceeds to execute the command by printing “Internet connection is active”, otherwise the tool wouldn’t execute the command.

3. Initial Setup and Cleanup:

The tool lists the configuration settings and flags used. The force flag (-f) is confirmed, indicating that previous output files are being cleaned up. For a flag used the output will print 1 otherwise it will print 0 and for the p-value/p-adjust threshold will portray the set threshold value.

4. Permission to run the analysis:

The tool takes confirmation from the user if they want to continue running the pipeline to continue the enrichment analysis. If the user writes “y”, “Y”, “yes”, “Yes” or “YES” then the tool will proceed to run the enrichment analysis otherwise if the user denies permission and writes “n”, “N”, “no”, “No”, or “NO” then the tool will not run the analysis.

5. Generating Plots:

- **Enrichment Analysis:**

The tool performs the enrichment analysis of KEGG pathways based on the provided genomic data and parameters. This step involves identifying significant pathways and calculating their enrichment scores. This also shows the exact time when this step is performed.

- **Generating General Plots:**

General plots are generated to provide an overview of the enrichment results. These might include bar plots or other summary visualisations. This also shows the exact time when this step is performed.

- **Generating PubMed Trend Plots:**

PubMed trend plots are generated for both P-adjust and P-value significant pathways. These plots show publication trends over time for the significant pathways. This also shows the exact time when this step is performed.

- **Generating KEGG Pathway Diagrams:**

KEGG pathway diagrams are generated to provide detailed visualisations of the significant pathways, highlighting the genes and their interactions. This also shows the exact time when this step is performed.

- **Generating Heatmap Plots:**

Heatmap plots are generated for both P-value and P-adjust significant pathways. These plots visualise the intensity of pathway significance across different conditions or datasets. This also shows the exact time when this step is performed.

- **Generating Upset Plots:**

Upset plots are generated for both P-value and P-adjust significant pathways. These plots show the intersections of significant pathways across different conditions or datasets. This also shows the exact time when this step is performed.

- **Generating Tree Plots:**

Tree plots are generated for both P-value and P-adjust significant pathways. These plots visualise the hierarchical relationships between significant pathways. This also shows the exact time when this step is performed.

6. Completion:

The tool completes the enrichment analysis and confirms successful execution. All specified plots are generated based on the analysis results.

- **Summary of Plots Generated (by this command)**

- **Bar Plots:** General visualisations of significant pathways.
- **BarPlot_qScore:** Visualisations of pathways based on adjusted p-values.
- **Dot Plots:** Dots representing pathways with significance and enrichment scores.
- **Lollipop Plots:** Hybrid of bar and dot plots for clear visualisation.
- **Cnet Plots:** Network visualisations of gene-pathway relationships.
- **Heatmap Plots:** Intensity maps of pathway significance.
- **Upset Plots:** Intersection visualisations of significant pathways.
- **Tree Plots:** Hierarchical relationships between significant pathways.
- **PubMed Trend Plots:** Publication trends for significant pathways.
- **KEGG Pathway Diagrams:** Detailed visualisations of biological pathways.

These plots provide a comprehensive view of the enrichment analysis results, aiding in the interpretation and understanding of the biological significance of the findings. All the plots are generated for both P-value and adjusted P-value.

- **How to view the enrichment analysis results?**

To view the results of the enrichment analysis, follow the commands in Figure 5. The user has to go back a directory using the command `cd ..` and then change the directory using the command `cd` and go to the directory named `out` to access the output files. The user can list all the files present in the current directory using the command `ls`. The result shows a CSV file named `"enrichment_KEGG_results.csv"` and two folders named `"graph"` containing all the plots and another named `"pathway"` containing two more folders named `"p_adjust_based_pathway"` and `"p_value_based_pathway"`. The user can again change the directory using the command `cd graph` and list all the files present using the command `ls` to access all the plots from the

enrichment analysis. Similarly, the user can also access the pathways from the above-mentioned two folders. Refer to Figure 8 to understand these steps clearly.

```

bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE$ ls
annotation_files  getENRICH  input  out
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE$ cd out/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ ls
enrichment_KEGG_results.csv  graph  pathway
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ cd graph/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/graph$ ls
p_adjust_bar_plot.html      padjust_loollipop_plot.png      p_value_dot_plot.html
p_adjust_bar_plot.png       padjust_pubMed_trend_plot.png    p_value_dot_plot.png
p_adjust_bar_plot_qscore.html padjust_Treemap.png              pvalue_heatmap_plot.png
p_adjust_bar_plot_qscore.png padjust_upset_plot.png           p_value_loollipop_plot.html
p_adjust_cnetplot_interactive.html p_value_bar_plot.html            p_value_loollipop_plot.png
p_adjust_dot_plot.html      p_value_bar_plot_qscore.html     pvalue_pubMed_trend_plot.png
p_adjust_dot_plot.png       p_value_bar_plot_qscore.png      pvalue_Treemap.png
padjust_heatmap_plot.png    pvalue_cnetplot_interactive.html pvalue_upset_plot.png
padjust_loollipop_plot.html
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/graph$ cd ..
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ ls
enrichment_KEGG_results.csv  graph  pathway
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out$ cd pathway/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway$ ls
p_adjust_based_pathway  p_value_based_pathway
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway$ cd p_adjust_based_pathway/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway/p_adjust_based_pathway$ ls
ko00480.pathview.png      ko00590.pathview.png      ko02010.pathview.png      ko04142.pathview.png      ko04976.pathview.png
ko00480.png               ko00590.png               ko02010.png               ko04142.png               ko04976.png
ko00480.xml               ko00590.xml               ko02010.xml               ko04142.xml               ko04976.xml
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway/p_adjust_based_pathway$ cd ..
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway$ ls
p_adjust_based_pathway  p_value_based_pathway
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway$ cd p_value_based_pathway/
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway/p_value_based_pathway$ ls
ko00360.pathview.png      ko00590.pathview.png      ko02010.pathview.png      ko04210.pathview.png      ko04621.pathview.png
ko00360.png               ko00590.png               ko02010.png               ko04210.png               ko04621.png
ko00360.xml               ko00590.xml               ko02010.xml               ko04210.xml               ko04621.xml
ko00480.pathview.png      ko00790.pathview.png      ko04139.pathview.png      ko04212.pathview.png      ko04976.pathview.png
ko00480.png               ko00790.png               ko04139.png               ko04212.png               ko04976.png
ko00480.xml               ko00790.xml               ko04139.xml               ko04212.xml               ko04976.xml
ko00561.pathview.png      ko00830.pathview.png      ko04142.pathview.png      ko04612.pathview.png      ko05323.pathview.png
ko00561.png               ko00830.png               ko04142.png               ko04612.png               ko05323.png
ko00561.xml               ko00830.xml               ko04142.xml               ko04612.xml               ko05323.xml
bajay@bajay-SS400TR-34:~/Ajay_Bhatia/EXAMPLE/out/pathway/p_value_based_pathway$ |

```

Figure 11 Screenshots of the steps to view the results of the enrichment analysis for example 2

3.5. TYPES OF PLOTS THAT CAN BE GENERATED USING GETENRICH

The getENRICH tool provides several types of plots for visualising enrichment analysis results. Each plot has two formats: an interactive HTML version and a static PNG version. Both formats can be viewed and downloaded by the user. Additionally, each plot is available in two versions: one based on p-value and another based on p-adjust. Below is a detailed explanation of each plot type:

i. Enrichment Analysis Bar Plot

Description: This plot visualises the enrichment analysis results as a bar graph. Each bar represents a pathway or a gene set, with the height of the bar corresponding to the enrichment score or significance level (p-value or adjusted p-value).

Usage: Helps in quickly identifying the most significant pathways or gene sets.

Example:

```
./getENRICH -c config.json -v
```

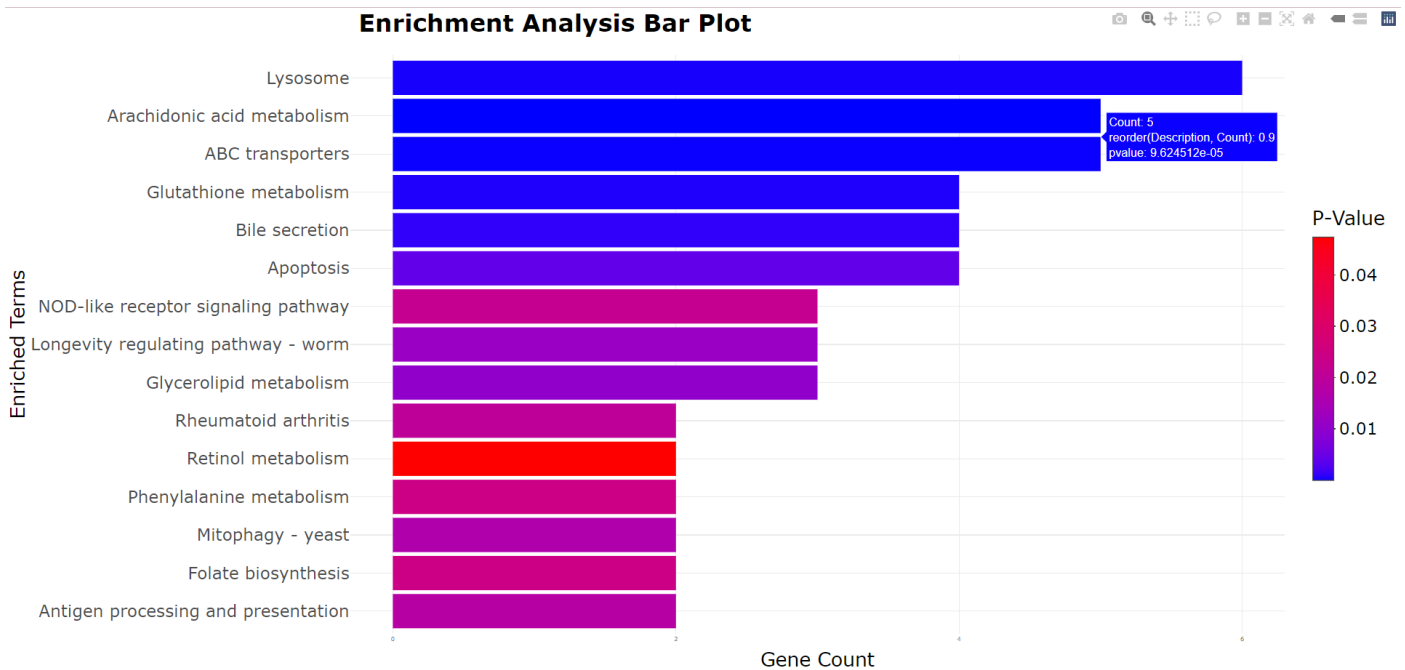


Figure 12 Enrichment Analysis Bar Plot

ii. Enrichment Analysis Bar Plot_qScore

Description: Similar to the standard bar plot, but the bars represent the q-score (adjusted p-value) instead of the raw p-value. This plot is useful for visualising the significance of pathways after adjusting for multiple testing.

Usage: Provides a more stringent view of pathway significance.

Example:

```
./getENRICH -c config.json -a
```

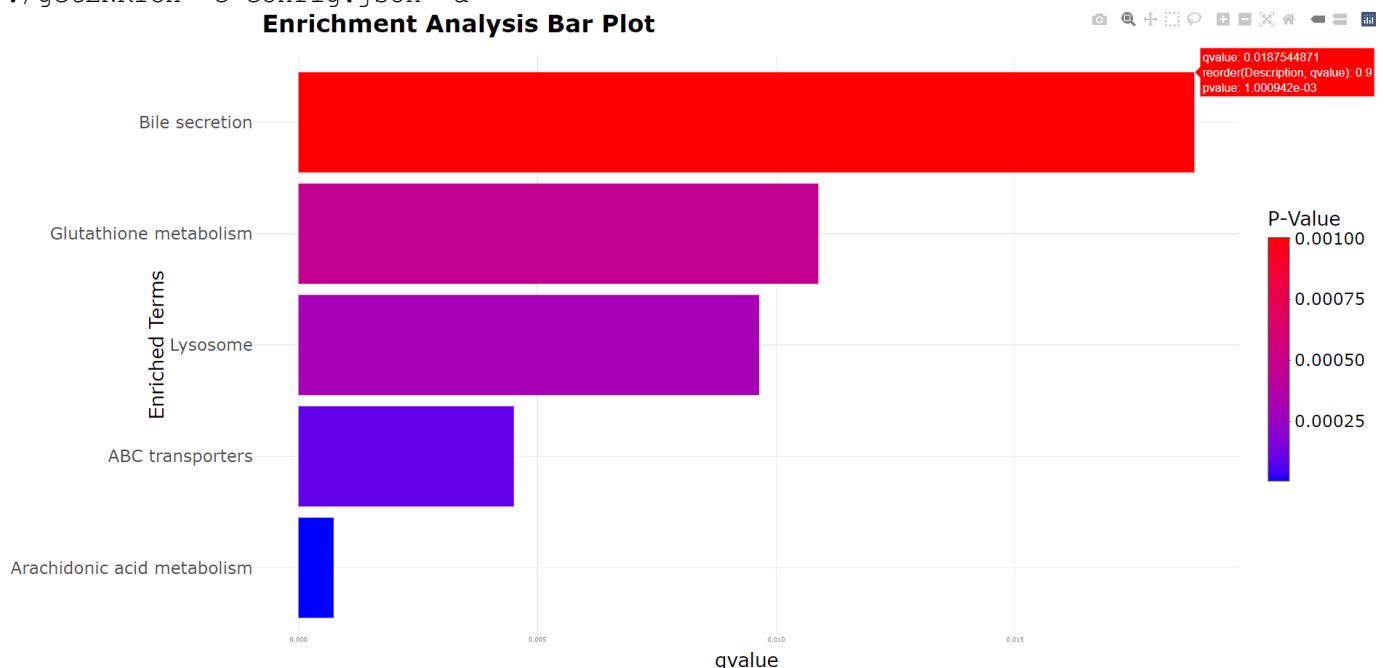


Figure 13 Enrichment Analysis Bar Plot_qScore

iii. Enrichment Analysis Dot Plot

Description: This plot uses dots to represent pathways or gene sets, with the position of the dot indicating the significance level (p-value or adjusted p-value) and the size of the dot indicating the enrichment score.

Usage: Allows for easy comparison of significance levels and enrichment scores simultaneously.

Example:

```
./getENRICH -c config.json -i
```

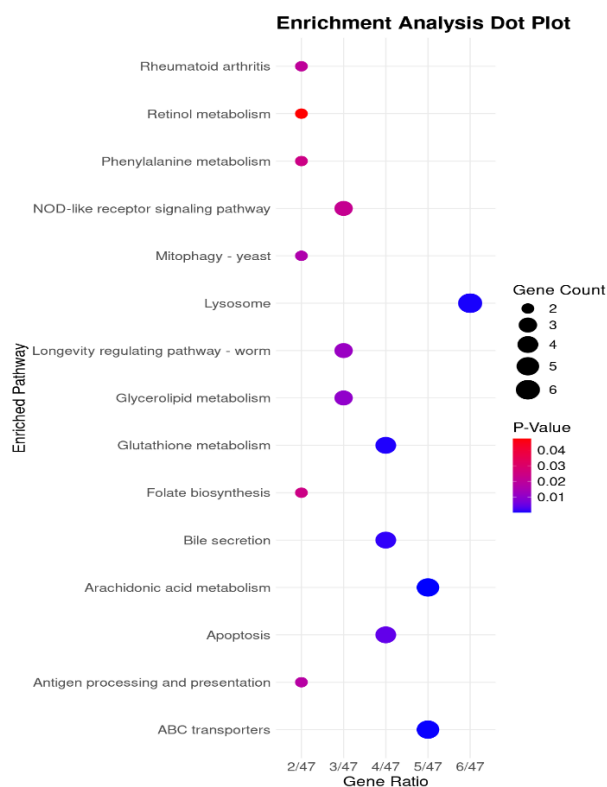


Figure 14 Enrichment Analysis Dot Plot

iv. Enrichment Analysis Lollipop Plot

Description: A hybrid between a bar plot and a dot plot, the lollipop plot shows a line with a dot at the end, where the length of the line indicates the significance level and the dot represents the enrichment score.

Usage: Combines the advantages of bar and dot plots for a clearer visual representation.

Example:

```
./getENRICH -c config.json -m
```

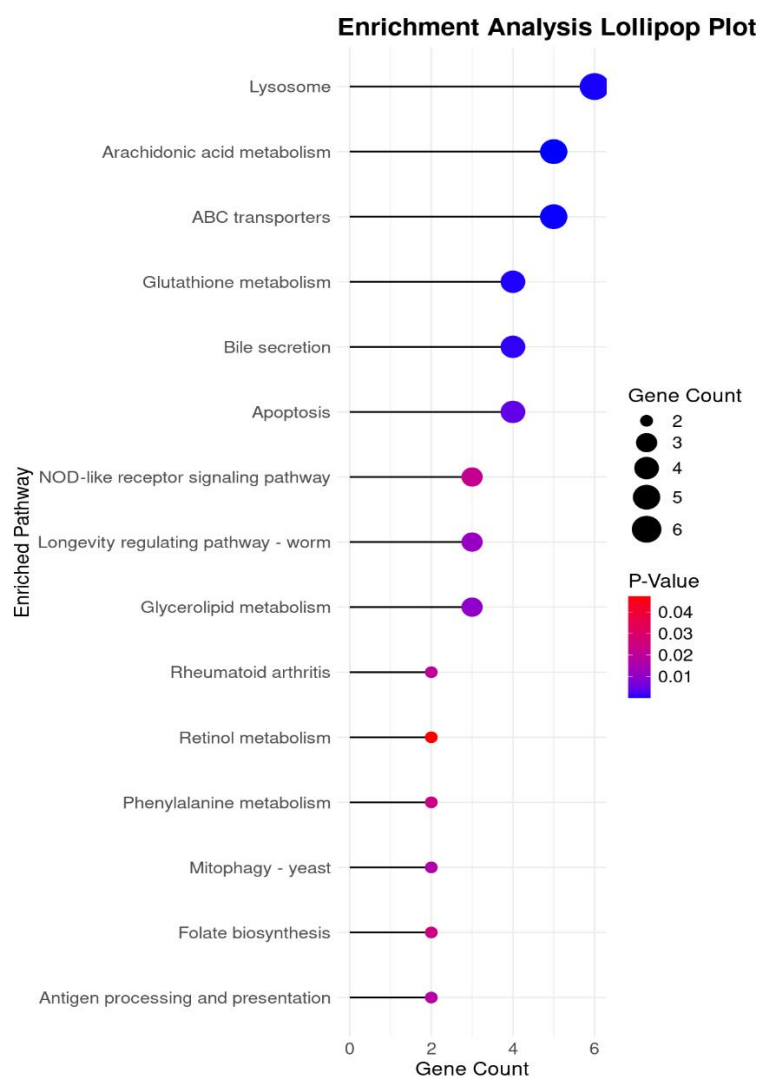



Figure 15 Enrichment Analysis Lollipop Plot

v. Enrichment Analysis Cnet Plot

Description: A cnet plot (circle network plot) visualises the relationships between different genes and pathways. Nodes (depicted in yellow) represent genes or pathways, and edges represent interactions or shared memberships, and the blue nodes represent linked genes.

Usage: Ideal for exploring the network structure and interconnections of significant pathways.

Example:

```
./getENRICH -c config.json -j
```

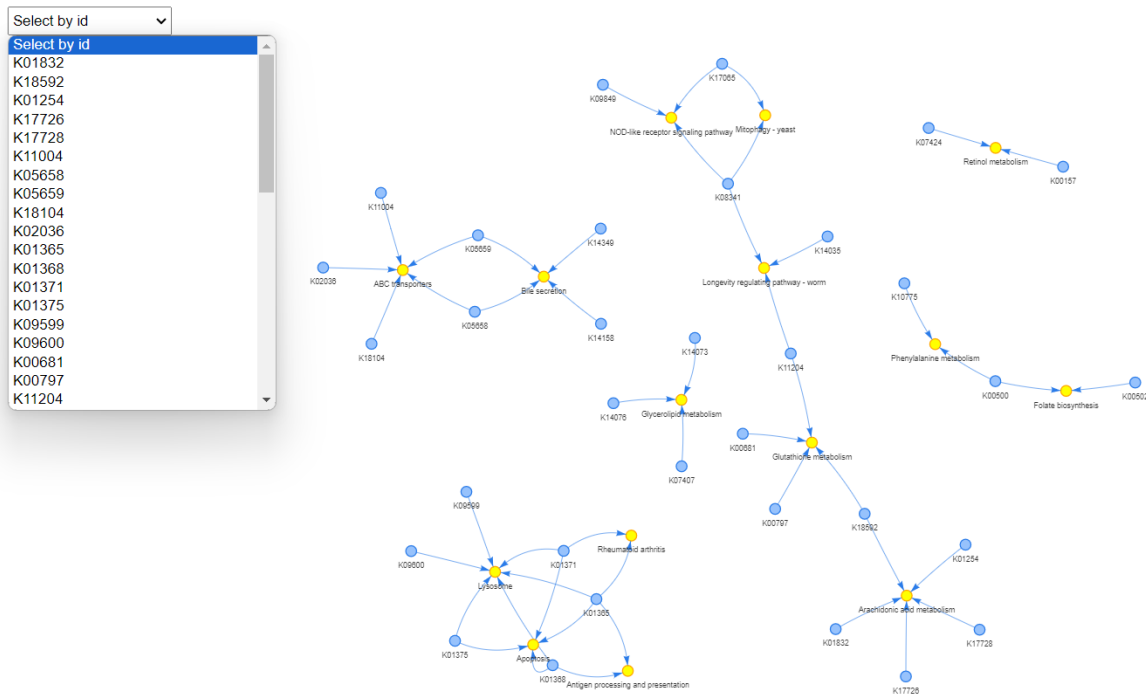


Figure 16 Enrichment Analysis interactive Cnet Plot

vi. Enrichment Analysis Heat map Plot

Description: This plot displays a heatmap of significant pathways, where colours represent the significance levels (p-values or adjusted p-values) and/or enrichment scores.

Usage: Useful for identifying clusters of significant pathways and comparing significance levels across conditions.

Example:

```
./getENRICH -c config.json -i
```

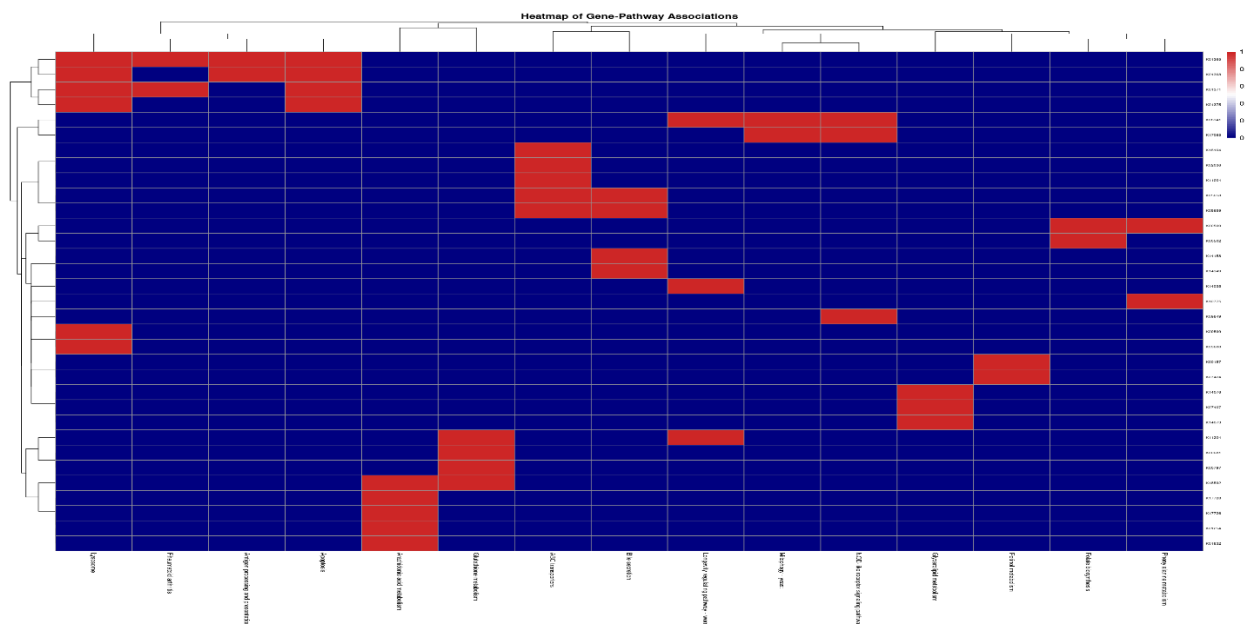


Figure 17 Enrichment Analysis Heat map Plot

vii. Enrichment Analysis Upset Plot

Description: An upset plot visualises the intersections of significant pathways across different conditions or datasets. It shows the size of each set and the intersections between them.

Usage: Helps in understanding the overlap and uniqueness of significant pathways between different analyses.

Example:

```
./getENRICH -c config.json -k
```

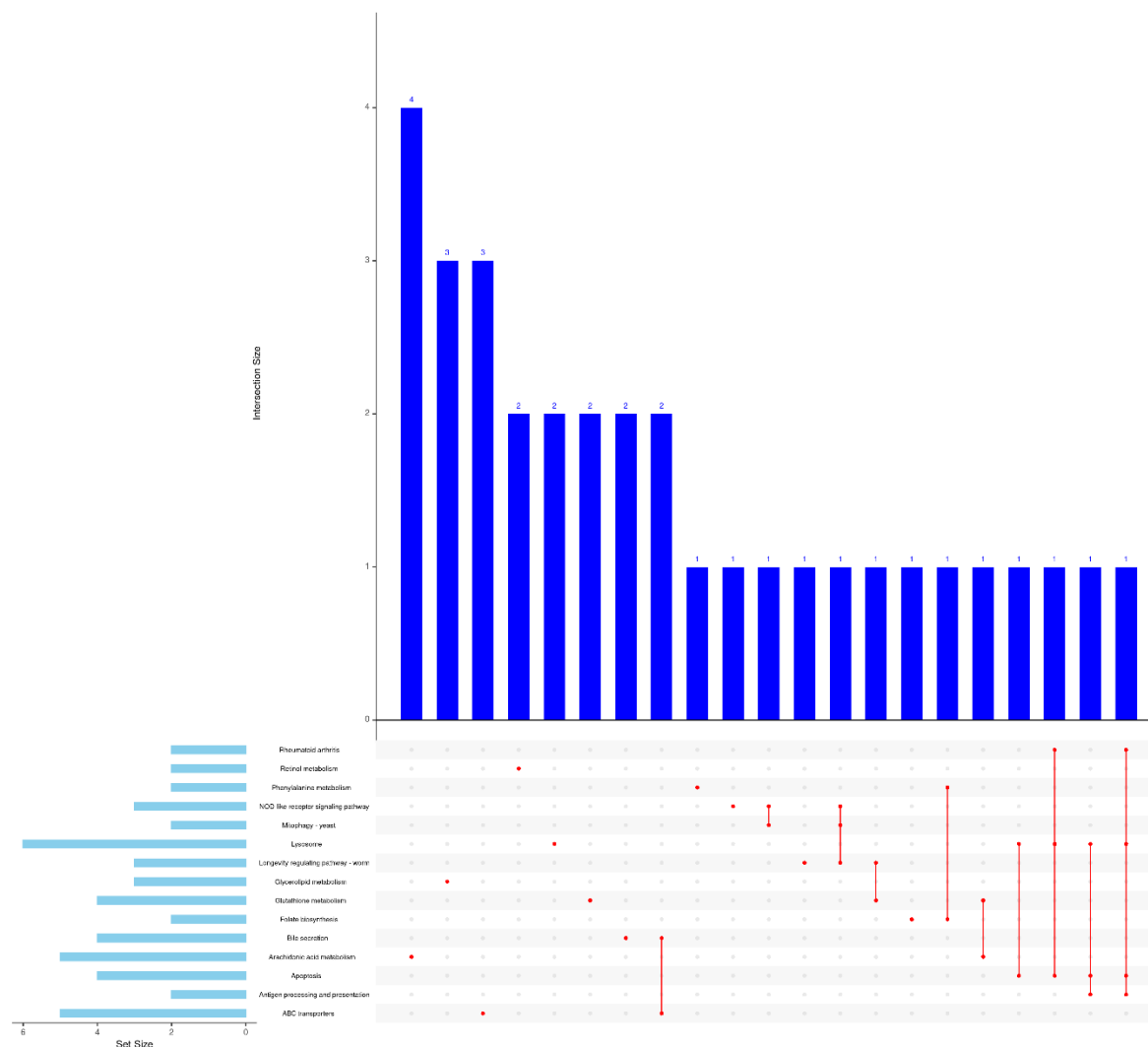


Figure 18 Enrichment Analysis Upset Plot

viii. Enrichment Analysis Tree Plot

Description: A tree plot visualizes the hierarchical relationships between significant pathways, often clustering similar pathways together.

Usage: Useful for understanding the hierarchical structure and grouping of pathways.

Example:

```
./getENRICH -c config.json -m
```

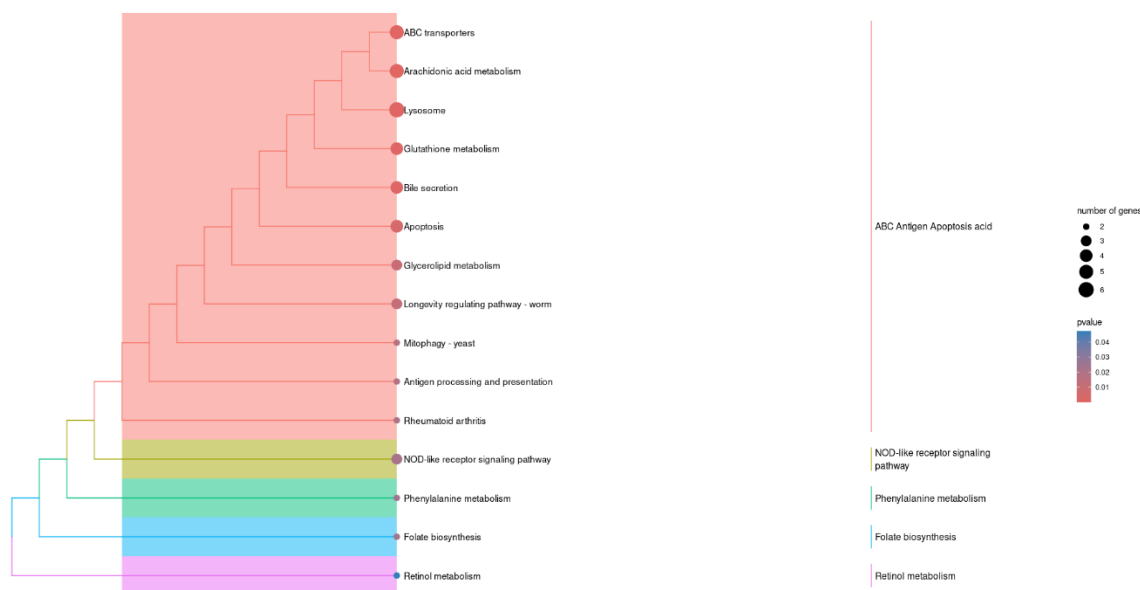


Figure 19 Enrichment Analysis Tree Plot

ix. Enrichment Analysis PubMed Trend Plot

Description: This plot shows the publication trends over time for significant pathways. It tracks the number of publications related to each pathway over a specified period.

Usage: Highlights the research interest and trends in specific pathways.

Example:

```
./getENRICH -c config.json -o
```

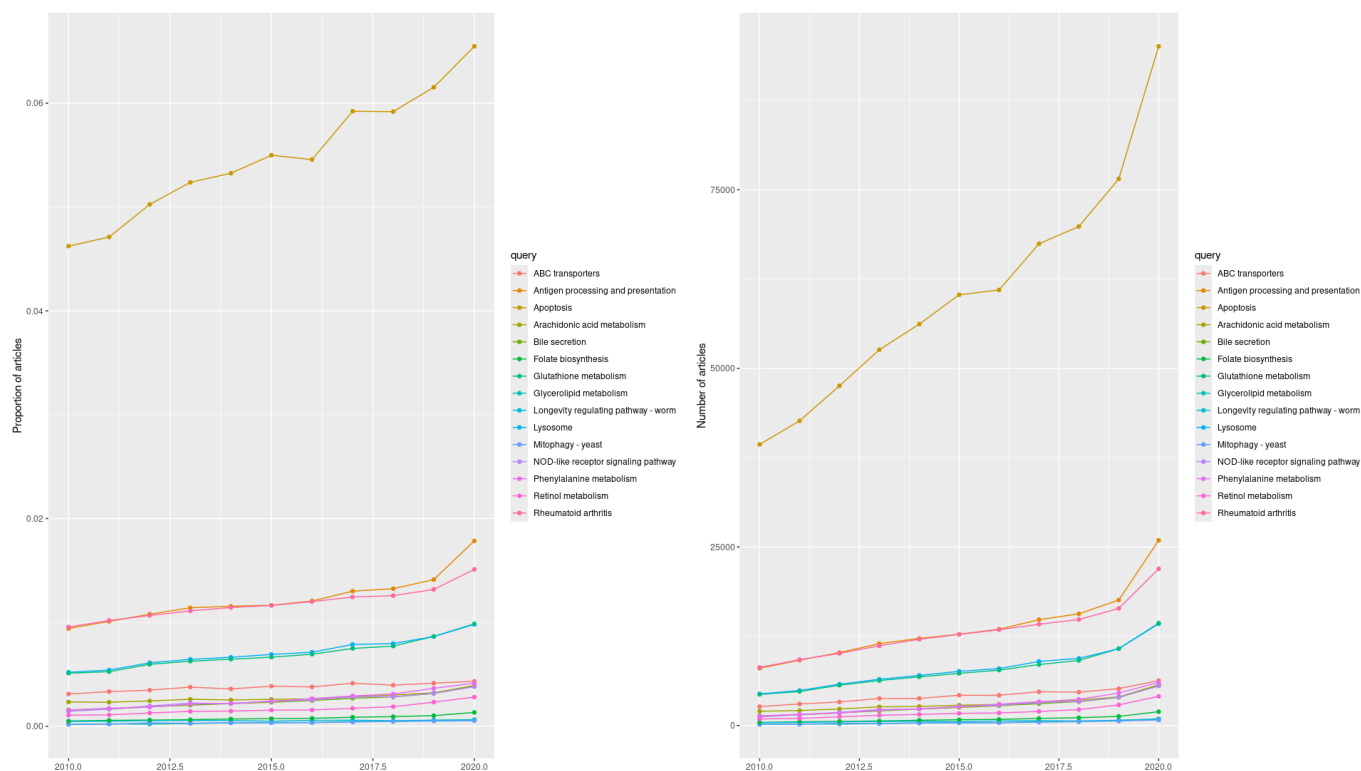


Figure 20 Enrichment Analysis PubMed Trend Plot

x. Enrichment Analysis KEGG Pathway Diagrams

Description: These diagrams provide detailed visualizations of KEGG pathways, highlighting the positions and interactions of genes within the pathway. They are annotated with significance information from the enrichment analysis.

Usage: Offers a comprehensive view of the biological pathways and their significance in the context of the analyzed data.

Example:

```
./getENRICH -c config.json -v
```

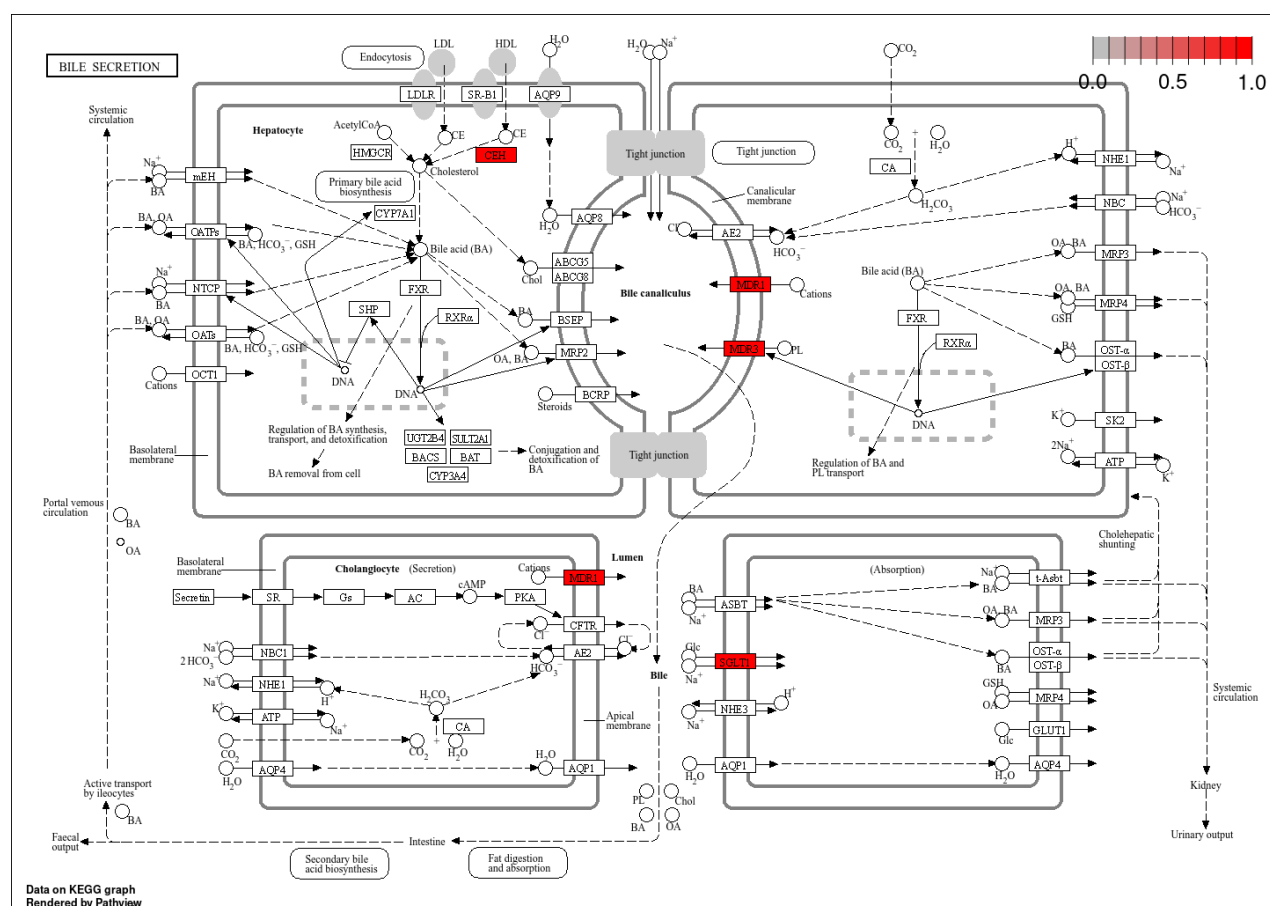


Figure 21 Data on KEGG graph showing bile secretion pathway

4. TROUBLESHOOTING

Common Errors

- **Error 1:** Description and solution.
- **Error 2:** Description and solution.

FAQ

- **Question 1:** Answer.
- **Question 2:** Answer.

References and Further Reading

- [Related Paper 1](#)
- [Related Paper 2](#)

Reporting Issues

Report bugs or request features via the [issue tracker](#).

Contact Information

Email

Support: support@getenrich-nm.org

Forums/Community

Join the community forum [here](#).

Issue Tracker

Submit issues on [GitHub](#).