**richPathR**

**Tutorial**

1. **Download and install the package**

This package can be downloaded from GitHub account: <https://github.com/niams-bdmds/richPathR> and installed locally or it can be installed directly from the GitHub.

*>install(devtools)*

*>Install\_github(“niams-bdmds/richPathR”*

(NOTE: currently the package in GitHub is set private. Install.github functionality does not work for now.)

========================================================

NOTE: in developing and testing stage, make sure all the packages are available in your machine including devtools and roxygen. Navigate to the directory, click build icon in Rstudio and click load\_all. All the functions are loaded. The test data is available in /data dir for testing the package. Tutorial makes navigation easy.

**=======================================================**

1. **Required packages and dependencies**

This package was written in MacBookPro 2019 and tested in Windows. Make sure the following packages are up to date in your computing environment. Install the following or later versions of the R packages.

1. dplyr(1.0.8)

2. enrichR(3.0)

3. filestrings(3.2.2)

4. ggplot2(3.3.5)

5. pheatmap(1.0.12)

6. plotly(4.10.0)

7. purrr(0.3.4)

5. readxl(1.4.0)

6. tidyr(1.2.0)

7. xlsx(0.6.5)

8. R(4.1.3)

The following versions of R development tools were used to write this package

9. devtools(2.4.3)

10. roxygen2(7.2.0)

The easiest way of calling the required package is using the following function. Run the following command:

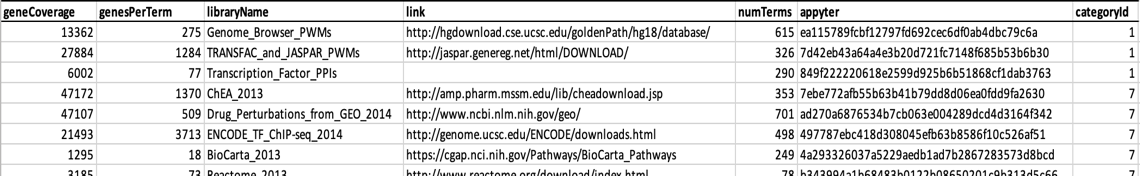
*>call\_required\_packages()*

1. **Obtaining the most recent database from *enrichr***

It is highly recommended to download the current database from *enrichr* web application. Use the following command to obtain the library database. It implements listEnrichrDbs functionality hosted in *enrichr* package.

>db\_EnrichR\_lib()

This command will yield a table something like this:

****

1. **Implementing *enrichr* and generating two data frames**

Put the csv file of gene set to be explored in /data dir. Any number of gene sets and libraries can be used as input for implementing this package and quickly obtaining large data frames and generate tables of the most common and unique *terms*. However, for visualizations purpose, we recommend a maximum of five gene sets and an equal number of libraries. First specify the list of database libraries of interest as follows.

*>dbs <- c("Cancer\_Cell\_Line\_Encyclopedia", "NCI-60\_Cancer\_Cell\_Lines", "NCI-Nature\_2016", "UK\_Biobank\_GWAS\_v1", "KEGG\_2021\_Human")*

Use the following command to implement *enrichr* plug in to obtain the excel files. The output of this command will generate /results directory in /data directory. For each gene list, the result directory contains a separate excel file. For each library, there is a separate sheet in the file.

*>implement\_enrichr(dbs=dbs, gene\_list = "data/prc\_gene\_set.csv")*

The following two commands will generate the data frames, the second command will generate the larger data frame.

*enrichr\_df <- enrichr\_df()*

*expanded\_enrichr\_df <- expanded\_enrichr\_df()*

1. **Visualization**

Obtaining the large data frame and generating the most unique and ubiquitous terms are very useful functions of this package. Once a data frame with multiple gene lists and libraries is obtained, the users can use their script for visualizations. The followings are the exploratory visualization tools provided in the package.

* 1. **Bar plots**

Use the following function to generate a bar plot of term count distributed across the gene list and libraries. The minimum\_combined score can be any positive integer, the default is 5. Make sure enrichr\_df generated by enrichr\_df() is available in the computing environment.

*bar\_plot\_count(enrichr\_df = enrichr\_df , minimum\_combined\_score )*

*Implementing this function generates the plot something like this.*

**Chart

Description automatically generated**

Use the following command to generate a mixed bar plot of the distribution of counts across gene sets and libraries. The minimum\_combined score can be any positive integer; the default is 5.

*>bar\_plot\_genelist\_library(enrichr\_df = enrichr\_df, minimum\_combined\_score = 5 )*

**Chart, bar chart

Description automatically generated**

Use the following command to visualize the top 30 (combined\_score sorted) terms distributed across the gene list. This visualization is particularly useful to see the unique and common most significant terms distributed. The minimum\_combined score can be any positive integer; the default is 5.

*>bar\_genelist\_terms(enrichr\_df = enrichr\_df, minimum\_combined\_score)*

**A picture containing chart

Description automatically generated**

* 1. **Violin plots**

Use the following function to visualize the log\_combined score as the violin plots. The combined score can be any positive integer, default is 5. This plot is extremely useful to scan the most significant terms across multiple gene lists and libraries.

*>violin\_plot\_genelist(enrichr\_df = enrichr\_df, minimum\_combined\_score )*

**Chart

Description automatically generated**

Use the following command to generate violin plots split into libraries.

>violin\_plot\_genelist\_library(enrichr\_df = enrichr\_df,

minimum\_combined\_score )

**Chart, bar chart

Description automatically generated**

* 1. **Heat map**

Use the following command to generate the heat map. Before using this functionality, generate expanded\_enrichr\_df and make it available in the computing environment. The minimum\_combined score can be any positive integer; the default is 5. This functionality breaks down genes from the top 30 most significant gene lists and visualizes the top hit genes distributed across gene lists and libraries.

*>enrichr\_heat\_map(expanded\_enrichr\_df = enrichr\_df, minimum\_combined\_score)*

**Chart

Description automatically generated**

* 1. **Tile plot**

To explore which gene lists hit most in which library, use the tile\_plot function. Before implementing this function, expanded *enrichr* data frame should be available in computing environment. The minimum\_combined score can be any positive integer; the default is 5.

*>tille\_plot(expanded\_enrichr\_df = expanded\_enrichr\_df, minimum\_combined\_score )*

**Table

Description automatically generated**

1. **Generating tables**

This is very useful function to explore the unique terms distributed across multiple gene list and libraries. Before implementing this function*, enrichr* data frame should be available in the computing environment. The combined score can be any positive integer, the default is 5. This table contains more rows than generating ubiquitous terms because all the terms with count one are listed in this table which is usually more than the max count.

*>enrichr\_unique\_terms(enrichr\_df = enrichr\_df, minimum\_combined\_score )*

Graphical user interface, application, table, Excel

Description automatically generated

This is a very useful function to explore the most ubiquitous terms distributed across multiple gene lists and libraries. Before implementing this function, *enrichr* data frame should be available in the computing environment. The combined score can be any positive integer, the default is 5.

The terms listed in the table are screened based on max(). Therefore, this table yields one or few rows.

*>enrichr\_ubiquitous\_terms(enrichr\_df = enrichr\_df, minimum\_combined\_score)*

Table

Description automatically generated