Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

diffEnrich: An R Package to Compare Functional Enrichment Between Two Experimentally-derived Groups of Genes by Connecting to the KEGG REST API

Harry A. Smith
Department of Biostatistics and Informatics
Colorado School of Public Health
Skaggs School of Pharmacy
and Pharmaceutical Sciences

Laura Saba
Skaggs School of Pharmacy
and Pharmaceutical Sciences

Abstract

Motivation: To aid in the biological interpretation of a list of candidate genes and proteins generated as part of omics studies, researchers quantitate the enrichment of known pathways or biological functions among the genes of interest. With the advent of new technologies and new experimental designs, it is often of interest to compare enrichment of a particular pathway between two gene lists (i.e., differential enrichment). **Results:** This package provides a number of functions that are intended to be used in a pipeline. Briefly, a function within the package will map species-specific ENTREZ gene IDs to their respective Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways by accessing the KEGG REST API. The KEGG API is used to guarantee the most up-to-date pathway data from KEGG. Next, another function will identify significantly enriched pathways in two gene sets independently. The user can then identify pathways that are differentially enriched between the two gene sets using a third function. This package also provides a plotting function. **Availability and implementation:** diffEnrich is freely available on the Comprehensive R Archive Network (CRAN). Issues and bug reports can be submitted to the GitHub page `https://github.com/SabaLab/diffEnrich/issues`. **Supplementary information:** A step-by-step tutorial is provided on the diffEnrich GitHub page `https://github.com/SabaLab/diffEnrich`, and example data are included in the package.

Keywords: differential enrichment, KEGG REST API, R.

1. Introduction

Often high throughput omics studies include a functional enrichment analysis to glean biological insight from a list of candidate genes, proteins, metabolites, etc. Functional enrichment examines whether the number of genes in the list associated with a biological function or particular pathway is more than would be expected by chance. As an example, enrichment of a particular pathway among a list of genes that are differentially expressed after an experimental manipulation may indicate that the pathway has been altered by that manipulation. This analysis is rather straight forward and many solutions have been offered (e.g., Huang *et al.* (2009); Kuleshov *et al.* (2016); Liao *et al.* (2019); Subramanian *et al.* (2005)). A wide variety of databases have also been used to define these pathways (e.g., Kanehisa and Goto (2000)) and ontologies (e.g., Ashburner *et al.* (2000)).

One key component of a statistically rigorous functional enrichment analysis is the definition of a background data set that can be used to estimate the number of candidate genes that are "expected" to be associated with the pathway by chance, e.g., if 5% of genes in the background data set are associated with a pathway then 5% of candidate gene are expected to be associated with the pathway by chance. For many study designs, the background data set is relatively simple to define (e.g., RNA-Seq analyses where the background data set includes genes expressed above background).

However, for some newer omics technologies, the background data set is hard to define. For example, LC-MS analysis can be used to identify carbonylated proteins (Petersen *et al.* (2018); Shearn *et al.* (2019); Shearn *et al.* (2018)). With this study design, carbonylated proteins are isolated using a BH-derivation and then LC-MS is used to identify peptides in this isolated sample. The most appropriate background data set would be proteins present in that tissue, but this would require a separate analytical analysis. Furthermore, most functional enrichment analyses involve a single gene list. However, in protein modification studies, the typical experimental design compares the presence or absence of particular modified proteins between multiple groups.

When there are two or more gene lists to compare and the background gene list is not clearly defined, as is often the case in protein modification experiments, we propose a differential enrichment analysis. In this analysis, we compare the proportion of genes/proteins from one gene list associated with a particular pathway to the proportion of genes/proteins from a second gene list that are associated with that pathway. To easily execute this analysis, we have designed an R package that uses the KEGG REST API to obtain the most recent version of the KEGG PATHWAY (Kanehisa and Goto (2000)) database to initially identify functional enrichment within a gene list using the entire KEGG transcriptome as the background data set and then to identify differentially enriched pathways between two gene lists. This R package includes a function to generate a "differential enrichment" graphic.

KEGG is a database resource for understanding high-level functions of a biological system, such as a cell, an organism and an ecosystem, from genomic and molecular-level information `https://www.kegg.jp/kegg/kegg1a.html`. KEGG is an integrated database resource consisting of eighteen databases that are clustered into 4 main categories: 1) systems information (e.g. hierarchies and maps), 2) genomic information (e.g. genes and proteins), 3) chemical information (e.g. biochemical reactions), and 4) health information (e.g. human disease and drugs) `https://www.kegg.jp/kegg/kegg1a.html`.

In 2012 KEGG released its first application programming interface (API), and has been adding

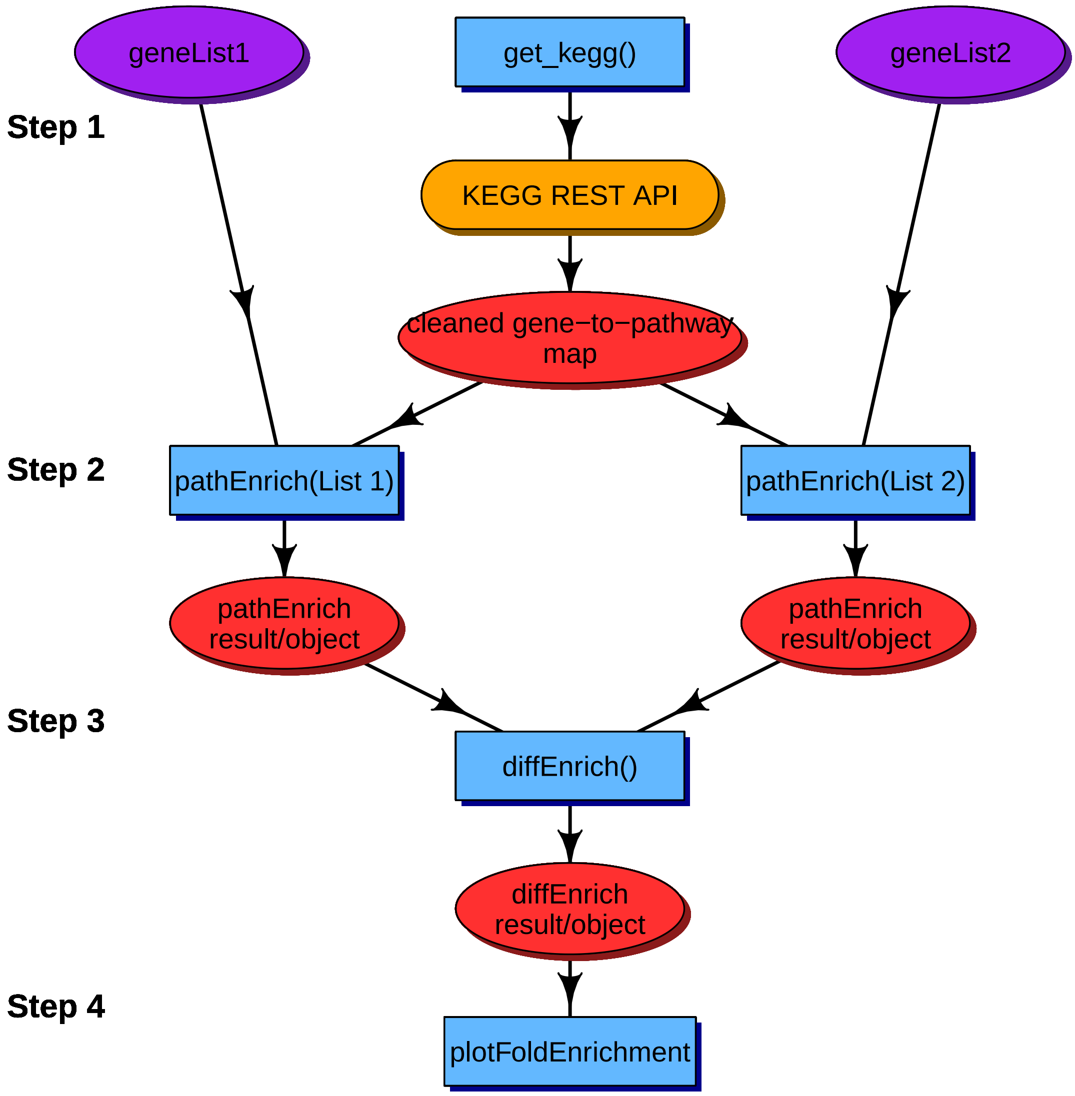

Figure 1: **diffEnrich Analysis pipeline.** Functions within the diffEnrich package are represented by blue rectangles. The data that must be provided by the user is represented by the purple ovals. Data objects generated by a function in diffEnrich are represented by red ovals. The external call of the `get_kegg` function to the KEGG REST API is represented in yellow.

features and functionality ever since. There are benefits to using an API. First, API's, like KEGG's, allow users to perform customized analyses with the most up-to-date versions of the data contained in the database. In addition, accessing the KEGG API is very easy using statistical programming tools like R or Python and integrating data retrieval into user's code makes the program reproducible. To further enforce reproducibilty diffEnrich adds a date and KEGG release tag to all data files it generates from accessing the API. For update histories and release notes for the KEGG REST API please visit `https://www.kegg.jp/kegg/rest/`.

2. Features

The goal of the diffEnrich package is to compare functional enrichment between two experimentally-derived groups of genes or proteins. This package provides four functions that are intended to be used in an ordered pipeline (Figure 1).

You can install the released version of diffEnrich from CRAN with:

```
install.packages("diffEnrich")
```

2.1. get_kegg: Download and prepare pathways from KEGG API

First, the `get_kegg` function is used to connect to the KEGG REST API and download the data sets required to perform downstream analysis. Currently, this function supports three species: Homo sapiens, Mus musculus, and Rattus norvegicus. For a given species, three data sets are generated: 1) ncbi_to_kegg: this data set maps NCBI/ENTREZ gene IDs to KEGG gene IDs, 2) kegg_to_pathway: this data set maps KEGG gene IDs to their respective KEGG pathway IDs, and 3) pathway_to_species: this data set maps KEGG pathway IDs to their respective pathway descriptions. This function typically completes in a few seconds, but it is important to note that the finishing time is dependent on the time it takes to connect to the KEGG API.

The `get__kegg` function accesses the KEGG REST API and downloads the data sets required to perform downstream analysis. This function takes two arguments. The first, 'species' is required. Currently, diffEnrich supports three species, and the argument is a character string using the KEGG code: Homo sapiens (human), use 'hsa'; Mus musculus (mouse), use 'mmu'; and Rattus norvegicus (rat), use 'rno'. The second, 'path' is also passed as a character string, and is the path of the directory in which the user would like to write the data sets downloaded from the KEGG REST API. If the user does not provide a path, the data sets will be automatically written to the current working directory using the `here::here()` (Müller (2017)) functionality. These data sets will be tab delimited files with a name describing the data, and for reproducibility, the date they were generated and the version of KEGG when the API was accessed. In addition to these flat files, `get_kegg` will also create a named list in R with the three relevant KEGG data sets. The names of this list will describe the data set, and are described in Table 1.

```
## Load package
suppressMessages(library(diffEnrich))
## run get_kegg() using rat
kegg_rno <- get_kegg('rno')

## 3 data sets will be written as tab delimited text files
## File location:  /Users/harry/Documents/Saba_Lab/diffEnrich
## Kegg Release:  Release_92.0+_11-22_Nov_19
```

Note: Because it is assumed that a user might want to use the data sets generated by `get_kegg`, it is careful not to overwrite data sets with exact names. `get_kegg` checks the path provided for data sets generated 'same-day/same-version', and if it finds even one of the three, it will not re-write any of the data sets. It will still however, let the user know it is not writing out new data sets and still generate the named list object. Users can generate 'same-day/same-version' data sets in different directories if they so choose.

```
## run get_kegg() using rat
kegg_rno <- get_kegg('rno')

## These files already exist in your working directory.  New files will not be
generated.
## Kegg Release:  Release_92.0+_11-22_Nov_19
```

Additionally, `get_kegg` can be used to read in saved versions of the txt files generated from a previous call, and generate an R list object that is compatible with downstream functions.

```
## run get_kegg() using rat
date <- as.character(Sys.Date())
kegg_rno <- get_kegg(read = TRUE,
                     path = here::here(),
                     date = date,
                     release = "92")

## Reading in the following files:
## ncbi_to_kegg2019-11-22Release_92.0+_11-22_Nov_19.txt
## kegg_to_pathway2019-11-22Release_92.0+_11-22_Nov_19.txt
## pathway_to_species2019-11-22Release_92.0+_11-22_Nov_19.txt
## File location:  /Users/harry/Documents/Saba_Lab/diffEnrich
```

Table 1: Description of the data sets retrieved by `get_kegg`'s connection to the KEGG REST API.

`get_kegg` list object	Description
ncbi_to_kegg	ncbi gene ID <– mapped to –> KEGG gene ID
kegg_to_pathway	KEGG gene ID <– mapped to –> KEGG pathway ID
pathway_to_species	KEGG pathway ID <– mapped to –> KEGG pathway species description

2.2. `pathEnrich`: Perform enrichment analysis of individual gene sets.

In this step, the `pathEnrich` function is used to identify KEGG pathways that are enriched (i.e. over-represented) based on a gene list of interest provided by the user. User gene lists must be ENTREZ gene IDs. If a user only has gene symbols, the `clusterProfiler` package (3.9) (Yu:2012) offers a function (`bitr`) that maps gene symbols and Ensembl IDs to ENTREZ gene IDs. An example of this function's use can be found in their vignette (`https://yulab-smu.github.io/clusterProfiler-book/chapter14.html#bitr`).

```
## View sample gene lists from package data
head(geneLists$list1)

## [1] "361692"    "293654"    "293655"    "500974"    "100361529"
## [6] "171434"

head(geneLists$list2)

## [1] "315547" "315548" "315549" "315550" "50938"  "58856"
```

The pathEnrich function will only use the genes from the list provided that are also in the KEGG database. The `pathEnrich` function should be run at least twice, once for the genes of interest in list 1 and once for the genes of interest in list 2. Each `pathEnrich` call generates a data frame summarizing the results of enrichment analyses in which a Fisher's Exact test is used to identify which KEGG pathways are enriched within the user's list of genes compared to all genes annotated to a KEGG pathway. Users can limit which pathways are tested by requiring that they contain a minimum number of genes from the list, and this can be set by changing the 'N' arguement. The default is that a KEGG pathway must contain at least 2 genes (N = 2) from the user's list to be tested.

By default, p-values from the Fisher's Exact test are adjusted for multiple comparisons with a False Discovery Rate (FDR) (Benjamini:1995), however users have the option of choosing any type of multiple testing correction supported by `p.adjust`. In addition to the unadjusted p-value and FDR, pathEnrich will calculate for each KEGG pathway, its fold enrichment defined as the ratio of number of genes observed from the gene list annotated to the KEGG pathway to the expected number of genes from the gene list to be annotated to the KEGG pathway by chance. An example of the first 6 results generated by `pathEnrich` are in Supplementary Table 1. For a detailed description of the variables in this table see Table 2.

Table 2: Description of columns is dataframe generated by pathEnrich.

Column Names	Column Description
KEGG_PATHWAY_ID	KEGG Pathway Identifier
KEGG_PATHWAY_description	Description of KEGG Pathway (provided by KEGG)
KEGG_PATHWAY_cnt	Number of Genes in KEGG Pathway
KEGG_PATHWAY_in_list	Number of Genes from gene list in KEGG Pathway

Table 2: Description of columns is dataframe generated by pathEnrich.

Column Names	Column Description
KEGG_DATABASE_cnt	Number of Genes in KEGG Database
KEGG_DATABASE_in_list	Number of Genes from gene list in KEGG Database
expected	Expected number of genes from list to be in KEGG pathway by chance
enrich_p	P-value for enrichment within the KEGG pathway for list genes
p_adj	Multiple testing adjusted enrichment p-values (default = False Discovery Rate (Benjamini and Hochberg, 1995))
fold_enrichment	Ratio of number of genes observed from the gene list annotated to the KEGG pathway to the number of genes expected from the gene list to be annotated to the KEGG pathway if there was no enrichment (i.e. KEGG_PATHWAY_in_list/expected)

S3 generic functions for `print` and `summary` are provided. The `print` function prints the results table as a `tibble`, and the `summary` function returns the number of pathways that reached statistical significance as well as their descriptions, the number of genes used from the KEGG data base, the KEGG species, and the method used for multiple testing correction. See supplementary information for examples.

2.3. `diffEnrich`: Identify differentially enriched KEGG pathways.

The `diffEnrich` function merges results from the `pathEnrich` function generated in section 2.2. This merged data set is then used to perform differential enrichment using a Fisher's exact test as described in 2.2. The resulting odds ratio is defined as the odds of a gene from list 2 belonging to a given KEGG pathway divided by the odds of a gene from list 1 belonging to a given KEGG pathway. Users have the same options for multiple testing methods that are provided in the `pathEnrich` function. KEGG pathways that do not contain any genes from either gene list (e.g. ‘rno04530’ contains 0 genes from list 1 and 0 genes from list 2) are removed from the analysis. If this is the case a warning will be printed that tells the user how many pathways were removed. This can be avoided by setting the ’N’ parameter to a value >0 in the `pathEnrich calls`. This diffEnrich function generates a table that contains the results from the analyses performed in section 2.2 for each gene list as well as odds ratios and their associated unadjusted and adjusted p-values for each KEGG pathway. For a detailed description of the results generated by `diffEnrich` see Table 3, and an example of the first 6 results generated by `diffEnrich` are in Supplementary Table 2.

```
## Perform differential enrichment
diff_enrich <- diffEnrich(list1_pe = list1_pe,
                          list2_pe = list2_pe,
                          method = 'none',
                          cutoff = 0.05)
```

The result of the `diffEnrich` call is a list object that contains a data frame with the estimated odds ratio generated by the Fisher’s Exact test and the associated p-value. S3 generic functions for `print` and `summary` are provided. The `print` function prints the results table as a `tibble`, and the `summary` function returns the number of pathways that reached statistical significance as well as their descriptions, the number of genes used from the KEGG database, the KEGG species, the number of pathways that were shared (i.e. had at least N gene from each gene list present in the pathway based on what the user chose for N in `pathEnrich`) between the gene lists and the method used for multiple testing correction.

3. Supplementary Information

Table 3: Supplementary Table 1. First 6 results generated by pathEnrich.

	KEGG_PATHWAY_ID	KEGG_PATHWAY_description	KEGG_PATHWAY_cnt	KEGG_PATHWAY_in_list	KEGG_DATABASE_cnt	KEG_DATABASE_in_list	expected	enrich_p	p_adj	fold_enrichment
95	rno04530	Tight junction - Rattus norvegicus (rat)	170	19.00	8856	295	5.66	0.00	0.00	3.36
172	rno05135	Yersinia infection - Rattus norvegicus (rat)	128	16.00	8856	295	4.26	0.00	0.00	3.75
194	rno05210	Colorectal cancer - Rattus norvegicus (rat)	88	12.00	8856	295	2.93	0.00	0.00	4.09
212	rno05231	Choline metabolism in cancer - Rattus norvegicus (rat)	99	12.00	8856	295	3.30	0.00	0.00	3.64
197	rno05213	Endometrial cancer - Rattus norvegicus (rat)	58	9.00	8856	295	1.93	0.00	0.00	4.66
66	rno04144	Endocytosis - Rattus norvegicus (rat)	275	22.00	8856	295	9.16	0.00	0.00	2.40

Example of S3 functions

```
summary(list1_pe)

## 219 KEGG pathways were tested.
##  Only KEGG pathways that contained at least 2 genes from gene_list were tested.
##  KEGG pathway species: Rattus norvegicus (rat)
##  8856 genes from gene_list were in the KEGG data pull.
##  p-value adjustment method: BH
##  36 pathways reached statistical significance after multiple testing correction at a cu
##
## Significant pathways:
##  Tight junction
## Yersinia infection
## Colorectal cancer
```

```
## Choline metabolism in cancer
## Endometrial cancer
## Endocytosis
## Neurotrophin signaling pathway
## Thermogenesis
## Oocyte meiosis
## VEGF signaling pathway
## Thyroid hormone signaling pathway
## Hippo signaling pathway
## T cell receptor signaling pathway
## Apoptosis
## Hepatocellular carcinoma
## MAPK signaling pathway
## Focal adhesion
## Salmonella infection
## Non-alcoholic fatty liver disease (NAFLD)
## ErbB signaling pathway
## Sphingolipid signaling pathway
## Pancreatic cancer
## Progesterone-mediated oocyte maturation
## Alzheimer disease
## Endocrine resistance
## Adrenergic signaling in cardiomyocytes
## IL-17 signaling pathway
## Chronic myeloid leukemia
## Dopaminergic synapse
## Prostate cancer
## EGFR tyrosine kinase inhibitor resistance
## Hepatitis C
## Ras signaling pathway
## Acute myeloid leukemia
## Insulin signaling pathway
## Fc epsilon RI signaling pathway
```

References

Ashburner, *et al.* (2000). "Gene ontology: tool for the unification of biology." *Nature genetics*, **25**(1), 25–29. doi:10.1038/75556.

Huang D, *et al.* (2009). "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists." *Nucleic acids research*, **37**(1), 1–13. doi:10.1093/nar/gkn923.

Kanehisa M, Goto S (2000). "KEGG: Kyoto Encyclopedia of Genes and Genomes." *Nucleic acids research*, **28**(1), 27–30. doi:10.1093/nar/28.1.27.

Kuleshov M, *et al.* (2016). "Enrichr: a comprehensive gene set enrichment analysis web server 2016 update." *Nucleic acids research*, **44**(1), 90–97. doi:10.1093/nar/gkw377.

Liao Y, *et al.* (2019). "Gene set analysis toolkit with revamped UIs and APIs." *Nucleic acids research*, **47**(1), 199–205. doi:10.1093/nar/gkz401.

Müller K (2017). *here: A Simpler Way to Find Your Files.* R package version 0.1, URL https://CRAN.R-project.org/package=here.

Petersen D, *et al.* (2018). "Elevated Nrf-2 responses are insufficient to mitigate protein carbonylation in hepatospecific PTEN deletion mice." *PLoS one*, **13**(5). doi:10.1371/journal.pone.0198139.

Shearn C, *et al.* (2018). "Knockout of the Gsta4 Gene in Male Mice Leads to an Altered Pattern of Hepatic Protein Carbonylation and Enhanced Inflammation Following Chronic Consumption of an Ethanol Diet." *Alcoholism clinical and experimental research*, **42**(7), 1192–1205. doi:10.1111/acer.13766.

Shearn C, *et al.* (2019). "Cholestatic liver disease results increased production of reactive aldehydes and an atypical periportal hepatic antioxidant response." *Free radical biology and medicine*, **143**(1), 101–114. doi:10.1016/j.freeradbiomed.2019.07.036.

Subramanian T, *et al.* (2005). "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles." *Proceedings of the National Academy of Sciences of the United States of America*, **102**(43), 15545–15550. doi:10.1073/pnas.0506580102.

Affiliation:

Laura Saba
University of Colorado
Skaggs School of Pharmacy and Pharmaceutical Sciences
Mail Stop C238
12850 E. Montview Blvd. V20-2124
Aurora, CO 80045
E-mail: Laura.Saba@cuanschutz.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd