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 (Peterson 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

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. Models and software

The basic Poisson regression model for count data is a special case of the GLM framework **?**. It describes the dependence of a count response variable y_i $(i = 1, \ldots, n)$ by assuming a Poisson distribution $y_i \sim \text{Pois}(\mu_i)$. The dependence of the conditional mean $\mathsf{E}[y_i \,|\, x_i] = \mu_i$ on the regressors x_i is then specified via a log link and a linear predictor

$$\log(\mu_i) \quad = \quad x_i^\top \beta, \tag{1}$$

where the regression coefficients β are estimated by maximum likelihood (ML) using the iterative weighted least squares (IWLS) algorithm.

> Note that around the `{equation}` above there should be no spaces (avoided in the LaTeX code by `%` lines) so that "normal" spacing is used and not a new paragraph started.

R provides a very flexible implementation of the general GLM framework in the function `glm()` (**?**) in the **stats** package. Its most important arguments are

```
glm(formula, data, subset, na.action, weights, offset,
  family = gaussian, start = NULL, control = glm.control(...),
  model = TRUE, y = TRUE, x = FALSE, ...)
```

where `formula` plus `data` is the now standard way of specifying regression relationships in R/S introduced in **?**. The remaining arguments in the first line (`subset`, `na.action`, `weights`, and `offset`) are also standard for setting up formula-based regression models in R/S. The arguments in the second line control aspects specific to GLMs while the arguments in the last line specify which components are returned in the fitted model object (of class '`glm`' which inherits from '`lm`'). For further arguments to `glm()` (including alternative specifications of starting values) see `?glm`. For estimating a Poisson model `family = poisson` has to be specified.

> As the synopsis above is a code listing that is not meant to be executed, one can use either the dedicated `{Code}` environment or a simple `{verbatim}` environment for this. Again, spaces before and after should be avoided.
>
> Finally, there might be a reference to a `{table}` such as Table 1. Usually, these are placed at the top of the page (`[t!]`), centered (`\centering`), with a caption below the table, column headers and captions in sentence style, and if possible avoiding vertical lines.

3. Illustrations

Type	Distribution	Method	Description
GLM	Poisson	ML	Poisson regression: classical GLM, estimated by maximum likelihood (ML)
		Quasi	"Quasi-Poisson regression": same mean function, estimated by quasi-ML (QML) or equivalently generalized estimating equations (GEE), inference adjustment via estimated dispersion parameter
		Adjusted	"Adjusted Poisson regression": same mean function, estimated by QML/GEE, inference adjustment via sandwich covariances
	NB	ML	NB regression: extended GLM, estimated by ML including additional shape parameter
Zero-augmented	Poisson	ML	Zero-inflated Poisson (ZIP), hurdle Poisson
	NB	ML	Zero-inflated NB (ZINB), hurdle NB

Table 1: Overview of various count regression models. The table is usually placed at the top of the page (`[t!]`), centered (`centering`), has a caption below the table, column headers and captions are in sentence style, and if possible vertical lines should be avoided.

For a simple illustration of basic Poisson and NB count regression the `quine` data from the **MASS** package is used. This provides the number of `Days` that children were absent from school in Australia in a particular year, along with several covariates that can be employed as regressors. The data can be loaded by

```
data("quine", package = "MASS")
```

and a basic frequency distribution of the response variable is displayed in Figure 1.

> For code input and output, the style files provide dedicated environments. Either the "agnostic" `{CodeInput}` and `{CodeOutput}` can be used or, equivalently, the environments `{Sinput}` and `{Soutput}` as produced by `Sweave()` or **knitr** when using the `render_sweave()` hook. Please make sure that all code is properly spaced, e.g., using `y = a + b * x` and *not* `y=a+b*x`. Moreover, code input should use "the usual" command prompt in the respective software system. For R code, the prompt `"R> "` should be used with `"+ "` as the continuation prompt. Generally, comments within the code chunks should be avoided – and made in the regular LaTeX text instead. Finally, empty lines before and after code input/output should be avoided (see above).

As a first model for the `quine` data, we fit the basic Poisson regression model. (Note that JSS prefers when the second line of code is indented by two spaces.)

```
m_pois <- glm(Days ~ (Eth + Sex + Age + Lrn)^2, data = quine,
  family = poisson)
```

To account for potential overdispersion we also consider a negative binomial GLM.

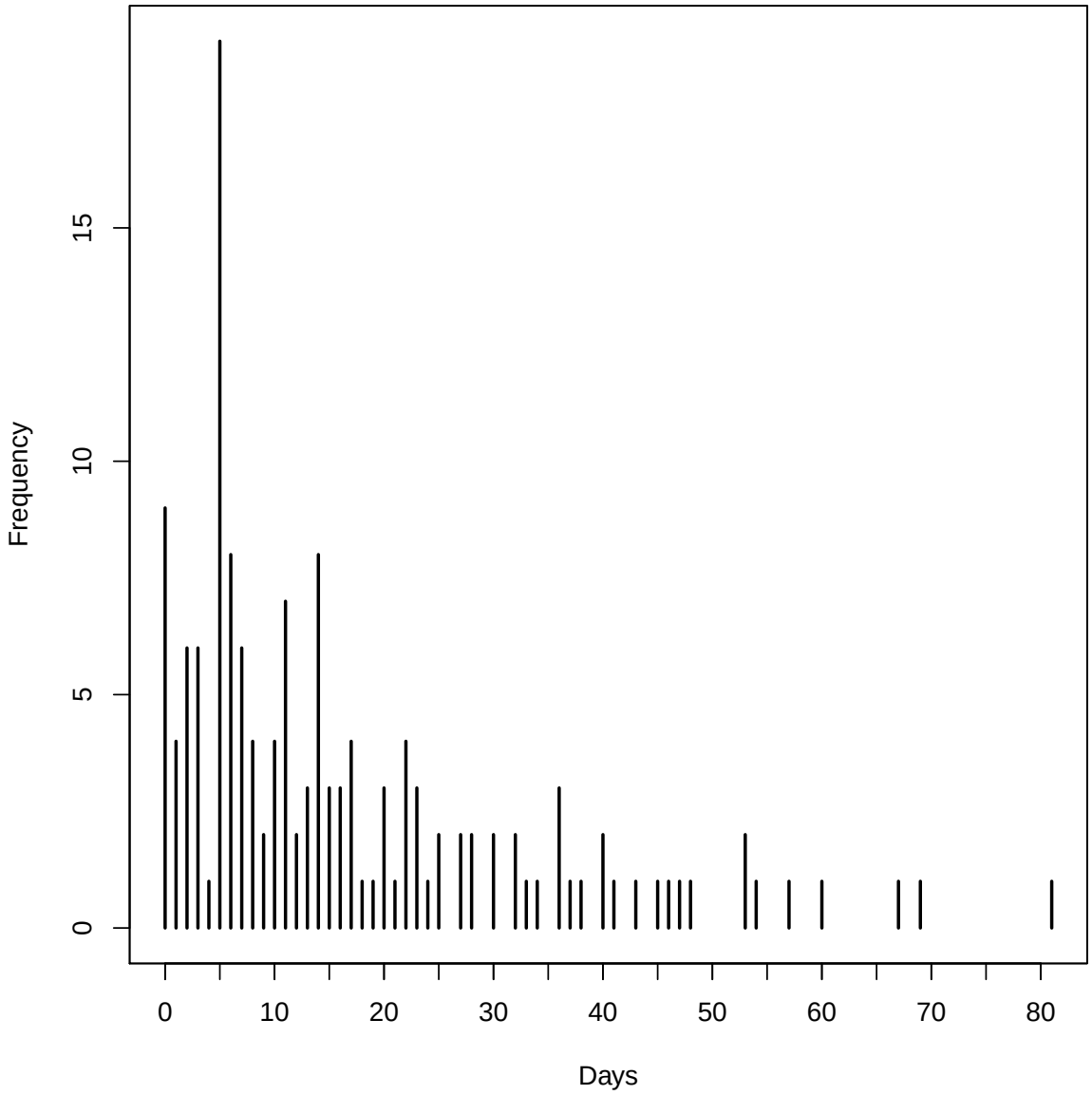

Figure 1: Frequency distribution for number of days absent from school.

```
library("MASS")
m_nbin <- glm.nb(Days ~ (Eth + Sex + Age + Lrn)^2, data = quine)
```

In a comparison with the BIC the latter model is clearly preferred.

```
BIC(m_pois, m_nbin)

##        df      BIC
## m_pois 18 2046.851
## m_nbin 19 1157.235
```

Hence, the full summary of that model is shown below.

```
summary(m_nbin)

##
## Call:
## glm.nb(formula = Days ~ (Eth + Sex + Age + Lrn)^2, data = quine,
##     init.theta = 1.60364105, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0857  -0.8306  -0.2620   0.4282   2.0898
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.00155    0.33709   8.904  < 2e-16 ***
## EthN        -0.24591    0.39135  -0.628  0.52977
## SexM        -0.77181    0.38021  -2.030  0.04236 *
## AgeF1       -0.02546    0.41615  -0.061  0.95121
## AgeF2       -0.54884    0.54393  -1.009  0.31296
## AgeF3       -0.25735    0.40558  -0.635  0.52574
## LrnSL        0.38919    0.48421   0.804  0.42153
## EthN:SexM    0.36240    0.29430   1.231  0.21818
## EthN:AgeF1  -0.70000    0.43646  -1.604  0.10876
## EthN:AgeF2  -1.23283    0.42962  -2.870  0.00411 **
## EthN:AgeF3   0.04721    0.44883   0.105  0.91622
## EthN:LrnSL   0.06847    0.34040   0.201  0.84059
## SexM:AgeF1   0.02257    0.47360   0.048  0.96198
## SexM:AgeF2   1.55330    0.51325   3.026  0.00247 **
## SexM:AgeF3   1.25227    0.45539   2.750  0.00596 **
## SexM:LrnSL   0.07187    0.40805   0.176  0.86019
## AgeF1:LrnSL -0.43101    0.47948  -0.899  0.36870
## AgeF2:LrnSL  0.52074    0.48567   1.072  0.28363
## AgeF3:LrnSL       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.6036) family taken to be 1)
##
##     Null deviance: 235.23  on 145  degrees of freedom
## Residual deviance: 167.53  on 128  degrees of freedom
## AIC: 1100.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.604
```

```
##           Std. Err.:  0.214
##
##  2 x log-likelihood:  -1062.546
```

4. Summary and discussion

- As usual ...

Computational details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section. Also, auxiliary packages (say, for visualizations, maps, tables, ...) that are not cited in the main text can be credited here.

The results in this paper were obtained using R 3.6.1 with the **MASS** 7.3.51.4 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

All acknowledgments (note the AE spelling) should be collected in this unnumbered section before the references. It may contain the usual information about funding and feedback from colleagues/reviewers/etc. Furthermore, information such as relative contributions of the authors may be added here (if any).

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.

A. More technical details

Appendices can be included after the bibliography (with a page break). Each section within the appendix should have a proper section title (rather than just *Appendix*).

For more technical style details, please check out JSS's style FAQ at https://www.jstatsoft.org/pages/view/style#frequently-asked-questions which includes the following topics:

- Title vs. sentence case.
- Graphics formatting.
- Naming conventions.
- Turning JSS manuscripts into R package vignettes.
- Trouble shooting.
- Many other potentially helpful details...

B. Using BibTeX

References need to be provided in a BibTeX file (`.bib`). All references should be made with `\cite`, `\citet`, `\citep`, `\citealp` etc. (and never hard-coded). This commands yield different formats of author-year citations and allow to include additional details (e.g., pages, chapters, ...) in brackets. In case you are not familiar with these commands see the JSS style FAQ for details.

Cleaning up BibTeX files is a somewhat tedious task – especially when acquiring the entries automatically from mixed online sources. However, it is important that informations are complete and presented in a consistent style to avoid confusions. JSS requires the following format.

- JSS-specific markup (`\proglang`, `\pkg`, `\code`) should be used in the references.
- Titles should be in title case.
- Journal titles should not be abbreviated and in title case.
- DOIs should be included where available.
- Software should be properly cited as well. For R packages `citation("pkgname")` typically provides a good starting point.

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