Package 'GSEAmining'

December 2, 2021

Type Package

Title Make Biological Sense of Gene Set Enrichment Analysis Outputs

Version 1.4.0

Description Gene Set Enrichment Analysis is a very powerful and interesting computational method that allows an easy correlation between differential expressed genes and biological processes. Unfortunately, although it was designed to help researchers to interpret gene expression data it can generate huge amounts of results whose biological meaning can be difficult to interpret.
Many available tools rely on the hierarchically structured Gene Ontology (GO) classification to reduce reundandcy in the results. However, due to the popularity of GSEA many more gene set collections, such as those in the Molecular Signatures Database are emerging. Since these collections are not organized as those in GO, their usage for GSEA do not always give a straightforward answer or, in other words, getting all the meaninful information can be challenging with the currently available tools. For these reasons, GSEAmining was born to be an easy tool to create reproducible reports to help researchers make biological sense of GSEA outputs.
Given the results of GSEA, GSEAmining clusters the different gene sets collections based on the presence of the same genes in the leadind edge (core) subset. Leading edge subsets are those genes that contribute most to the enrichment score of each collection of genes or gene sets. For this reason, gene sets that participate in similar biological processes should share genes in common and in turn cluster together. After that, GSEAmining is able to identify and represent for each cluster:
- The most enriched terms in the names of gene sets (as wordclouds)
- The most enriched genes in the leading edge subsets (as bar plots).
In each case, positive and negative enrichments are shown in different colors so it is easy to distinguish biological processes or genes that may be of interest in that particular study.

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports dplyr, tidytext, dendextend, tibble, ggplot2, ggwordcloud, stringr, gridExtra, rlang, grDevices, graphics, stats, methods

Depends R (>= 4.0)

Suggests knitr, rmarkdown, BiocStyle, clusterProfiler, testthat

VignetteBuilder knitr

biocViews GeneSetEnrichment, Clustering, Visualization

git_url https://git.bioconductor.org/packages/GSEAmining

git_branch RELEASE_3_14

git_last_commit 19dee20

git_last_commit_date 2021-10-26

Date/Publication 2021-12-02

Author Oriol Arqués [aut, cre]

Maintainer Oriol Arqués <oriol.arques@gmail.com>

R topics documented:

`clust_groups` *clust_groups*

Description

Takes the output of gm_clust, which is an hclust class object, and returns a data frame that will be used in the rest of GSEAmining functions gm_enrichreport, gm_enrichterms and gm_enrichcores.

Usage

```
clust_groups(df, hc)
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.
hc	The output of gm_clust, which is an hclust class object.

Value

A data.frame containing the cluster each gene set belongs to.

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
clust.groups <- clust_groups(genesets_sel, gs.cl)
```

---

clust_group_cores *clust_group_cores*

Description

Takes the output of clust_groups, a data frame , and process it to obtain the enrichment of genes in the core enrichment (or leading edge analysis) within each cluster. The output is used in the functions gm_enrichcores and gm_enrichreport.

Usage

```
clust_group_cores(cg, top = 3)
```

Arguments

cg	A data frame output from the GSEAmining clusts_groups function.
top	An integer to choose the top most enriched genes to plot per cluster. The default parameter are the top 3.

Value

A tibble with four variables (Cluster, Enrichment, lead_token, n).

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
clust.groups <- clust_groups(genesets_sel, gs.cl)
clust.lead <- clust_group_cores(clust.groups, top = 3)
```

clust_group_terms *clust_group_terms*

Description

Takes the output of clust_groups, a data frame , and process it to obtain the enrichment of terms in gene sets names within each cluster. The output is used in the functions gm_enrichterms and gm_enrichreport.

Usage

```
clust_group_terms(cg)
```

Arguments

cg	A data frame output from the GSEAmining cluster_groups function.

Value

A tibble with four variables (Cluster, Enrichment, monogram, n).

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
clust.groups <- clust_groups(genesets_sel, gs.cl)
clust.groups.wordcloud <- clust_group_terms(clust.groups)
```

genesets_sel *Selected gene sets as test*

Description

Data that corresponds to GSEA analysis of differential expressed genes from treated versus control samples in HGPalmer-PDX-P30 experiment. Differential gene expression was obtained by using the oligo and limma R packages. GSEA analysis was performed using the clusterProfiler R package using MSigDb collections C2, C5 and Hallmarks.

Usage

```
data(genesets_sel)
```

Format

An object of class data.frame with 52 observations and 4 variables:

ID Name of the gene set

NES Normalized Enrichment Score

p.adjust False discovery rate

core_enrichment Genes that are in the leading edge subset

Source

ArrayExpress

References

Arqués et al. Clinical Cancer Research. 2016 Feb 1;22(3):644-56. doi: 10.1158/1078-0432.CCR-14-3081. Epub 2015 Jul 29. (Clinical Cancer Research)

Examples

```
data(genesets_sel)

gs.cl <- gm_clust(genesets_sel)

gm_dendplot(genesets_sel, gs.cl)

gm_enrichterms(genesets_sel, gs.cl)

gm_enrichcores(genesets_sel, gs.cl)

## Not run: gm_enrichreport(genesets_sel, gs.cl)
```

---

gm_clust *gm_clust: GSEAmining cluster object*

Description

Takes the output of gm_filter or a data frame that with the results of GSEA analysis and returns and hclust object that can be plotted using the gm_dendplot function.

Usage

```
gm_clust(df)
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.

Value

An object of class hclust that contains the clustering of the gene sets by the core enriched genes.First a distance matrix is calculated using the 'binary' method and then a cluster with the 'complete' method is created.

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
```

---

gm_dendplot *gm_dendplot: GSEAmining dendrogram plotter*

Description

Takes the output of gm_clust, which is an hclust class object, and plots the dendrogram using the dendextend package.

Usage

```
gm_dendplot(
  df,
  hc,
  col_pos = "red",
  col_neg = "blue",
  dend_len = 30,
  rect = TRUE,
  rect_len = 2
)
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.
hc	The output of gm_clust, which is an hclust class object.
col_pos	Color to represent the positively enriched gene sets. Default is red.

col_neg	Color to represent the negatively enriched gene sets. Default is blue.
dend_len	An integer that defines the length of the dendrogram. Default value is 30. The closest to zero the longest the dendrogram.
rect	A logical value indicating if rectangles should be drawn around the clusters to help differentiating them. By default it is set to TRUE.
rect_len	An integer to specify the length of the rectangle around the cluster and the gene set label. Default is 2. The closest to zero the smallest the rectangle.

Value

Invisibly returns a list with all the elements necessary to plot a dendrogram.

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
gm_dendplot(genesets_sel, gs.cl)
```

---

gm_enrichcores *gm_enrichcores: GSEAmining core enrichment genes*

Description

Takes the output of gm_clust, which is an hclust class object, and plots the top n genes in core enrichment (leading edge analysis). Two options are available, either separate barplots by clusters or all together in one plot.

Usage

```
gm_enrichcores(
  df,
  hc,
  clust = TRUE,
  col_pos = "red",
  col_neg = "blue",
  top = 3
)
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.
hc	The output of gm_clust, which is an hclust class object.

clust	A logical value indicating if wordclouds should be separated by clusters or not. Default value is TRUE.
col_pos	Color to represent positively enriched gene sets. Default is red.
col_neg	Color to represent negatively enriched gene sets. Default is blue.
top	An integer to choose the top most enriched genes to plot per cluster. The default parameter are the top 3.

Value

Returns a ggplot object.

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
gm_enrichcores(genesets_sel, gs.cl)
```

---

gm_enrichreport *gm_enrichreport: GSEAmining Enrichment Report*

Description

Takes the output of gm_clust, which is an hclust class object, and creates a report in pdf that contains enriched terms and enriched core genes in gene sets for each cluster. The results of each cluster are plotted in an independent page.

Usage

```
gm_enrichreport(
  df,
  hc,
  col_pos = "red",
  col_neg = "blue",
  top = 3,
  output = "gm_report"
)
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.
hc	The output of gm_clust, which is an hclust class object.
col_pos	Color to represent positively enriched gene sets. Default is red.

col_neg	Color to represent negatively enriched gene sets. Default is blue.
top	An integer to choose the top most enriched genes to plot per cluster. The default parameter are the top 3.
output	A string to name the output pdf file.

Value

Generates a pdf file.

Examples

```
#' data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
## Not run: gm_enrichreport(genesets_sel, gs.cl)
```

---

gm_enrichterms *gm_enrichterms: GSEAmining enriched terms*

Description

Takes the output of gm_clust, which is an hclust class object, and plots gene set enriched terms as wordclouds. Two options are available, either separate enrichments by clusters or plot them together in a single plot.

Usage

```
gm_enrichterms(df, hc, clust = TRUE, col_pos = "red", col_neg = "blue")
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.
hc	The output of gm_clust, which is an hclust class object.
clust	A logical value indicating if wordclouds should be separated by clusters or not. Default value is TRUE.
col_pos	Color to represent positively enriched gene sets. Default is red.
col_neg	Color to represent negatively enriched gene sets. Default is blue.

Value

Returns a ggplot object.

Examples

```
data(genesets_sel)
gs.cl <- gm_clust(genesets_sel)
gm_enrichterms(genesets_sel, gs.cl)
```

---

gm_filter *gm_filter: GSEAmining GSEA output filter*

Description

Filters a data frame containing the results of GSEA analysis.

Usage

```
gm_filter(df, p.adj = 0.05, neg_NES = 1, pos_NES = 1)
```

Arguments

df	Data frame that contains at least three columns: an ID column for the gene set names, a NES column with the normalized enrichment score and a core_enrichment column containing the genes in the leading edge of each gene set separated by '/'.
p.adj	An integer to set the limit of the adjusted p-value (or false discovery rate, FDR). Default value is 0.05
neg_NES	A positive integer to set the limit of negative NES. Default is 1.
pos_NES	A positive integer to set the limit of positive NES. Default is 1.

Value

A data frame.

Examples

```
data(genesets_sel)
gs.filt <- gm_filter(genesets_sel, p.adj = 0.05, neg_NES = 2.6, pos_NES = 2)
```

stop_words	*Stop words. Eliminates the first word of the gene sets from MSigDb that relate to the origin of the gene set. Additionally it eliminates words that do not add a lot of significance such as prepositions or adverbs among others.*

Description

Stop words. Eliminates the first word of the gene sets from MSigDb that relate to the origin of the gene set. Additionally it eliminates words that do not add a lot of significance such as prepositions or adverbs among others.

Usage

```
stop_words()
```

Value

Returns a tibble with 2 variables.

Index