Package 'EGSEA'

November 11, 2026

Title Ensemble of Gene Set Enrichment Analyses

Version 1.40.0

Description This package implements the Ensemble of Gene Set Enrichment Analyses (EGSEA) method for gene set testing. EGSEA algorithm utilizes the analysis results of twelve prominent GSE algorithms in the literature to calculate collective significance scores for each gene set.

biocViews ImmunoOncology, DifferentialExpression, GO, GeneExpression, GeneSetEnrichment, Genetics, Microarray, MultipleComparison, OneChannel, Pathways, RNASeq, Sequencing, Software, SystemsBiology, TwoChannel,Metabolomics, Proteomics, KEGG, GraphAndNetwork, GeneSignaling, GeneTarget, NetworkEnrichment, Network, Classification

Depends R (>= 4.3.0), Biobase, gage (>= 2.14.4), AnnotationDbi, topGO (>= 2.16.0), pathview (>= 1.4.2)

Imports PADOG (>= 1.6.0), GSVA (>= 1.12.0), globaltest (>= 5.18.0), limma (>= 3.20.9), edgeR (>= 3.6.8), HTMLUtils (>= 0.1.5), hwriter (>= 1.2.2), gplots (>= 2.14.2), ggplot2 (>= 1.0.0), safe (>= 3.4.0), stringi (>= 0.5.0), parallel, stats, metap, grDevices, graphics, utils, org.Hs.eg.db, org.Mm.eg.db, org.Rn.eg.db, RColorBrewer, methods, EGSEAdata (>= 1.3.1), htmlwidgets, plotly, DT

License GPL-3

LazyLoad yes

NeedsCompilation no

Encoding UTF-8

Suggests BiocStyle, knitr, testthat

VignetteBuilder knitr

RoxygenNote 7.2.3

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

git_branch RELEASE_3_23

git_last_commit 1fc4087

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Monther Alhamdoosh [aut, cre],
Luyi Tian [aut],
Milica Ng [aut],
Matthew Ritchie [ctb]

Maintainer Monther Alhamdoosh <m.hamdoosh@gmail.com>

Contents

EGSEA-package 2
arraydata 3
buildIdx 3
egsea 8
egsea.sort 16
EGSEAResults 18
GSCollectionIndex 28

Index **31**

`EGSEA-package` *Ensemble of Gene Enrichment Analysis (EGSEA)*

Description

This packages provides the implementation of the EGSEA algorithm and addition functions to help perform GSE analysis.

Author(s)

Monther Alhamdoosh, Milica Ng and Matthew Ritchie

References

Monther Alhamdoosh, Milica Ng, Nicholas J. Wilson, Julie M. Sheridan, Huy Huynh, Michael J. Wilson, Matthew E. Ritchie; Combining multiple tools outperforms individual methods in gene set enrichment analyses. Bioinformatics 2017; 33 (3): 414-424. doi: 10.1093/bioinformatics/btw623

arraydata *Example dataset for* egsea.ma

Description

This is dataset is provided as an example only, for the egsea.ma function.

Usage

```
arraydata
```

Format

A Named List containing two elements, arrays and targets

arrays is a limma::EList with 4 values: source, E, genes and other.

targets is a data frame with 12 rows and 6 variables: Array, SampleID, Condition, Chip, Section and Experiment.

Source

URL

buildIdx *Functions to create gene set collection indexes for EGSEA*

Description

buildIdx indexes the MSigDB, KEGG and GeneSetDB collections to be used for the EGSEA analysis.

buildKEGGIdx prepares the KEGG pathway collection to be used for the EGSEA analysis.

buildMSigDBIdx prepares the MSigDB gene set collections to be used for the EGSEA analysis.

buildGeneSetDBIdx prepares the GeneSetDB gene set collections to be used for the EGSEA analysis.

buildCustomIdx creates a gene set collection from a given list of gene sets to be used for the EGSEA analysis.

buildGMTIdx creates a gene set collection from a given GMT file to be used for the EGSEA analysis.

Usage

```
buildIdx(
  entrezIDs,
  species = "human",
  msigdb.gsets = "all",
  gsdb.gsets = "none",
  go.part = FALSE,
  kegg.updated = FALSE,
  kegg.exclude = c(),
  min.size = 1
)

buildKEGGIdx(
  entrezIDs,
  species = "human",
  min.size = 1,
  updated = FALSE,
  exclude = c()
)

buildMSigDBIdx(
  entrezIDs,
  species = "Homo sapiens",
  geneSets = "all",
  go.part = FALSE,
  min.size = 1
)

buildGeneSetDBIdx(
  entrezIDs,
  species,
  geneSets = "all",
  go.part = FALSE,
  min.size = 1
)

buildCustomIdx(
  geneIDs,
  gsets,
  anno = NULL,
  label = "custom",
  name = "User-Defined Gene Sets",
  species = "Human",
  min.size = 1
)

buildGMTIdx(
  geneIDs,
```

```
  gmt.file,
  anno.cols = 0,
  anno.col.names = NULL,
  label = "gmtcustom",
  name = "User-Defined GMT Gene Sets",
  species = "Human",
  min.size = 1
)
```

Arguments

entrezIDs	character, a vector that stores the Entrez Gene IDs tagged in your dataset. The order of the Entrez Gene IDs should match those of the count/expression matrix row names.
species	character, determine the organism of selected gene sets: "human", "mouse" or "rat".
msigdb.gsets	character, a vector determines which gene set collections should be used from MSigDB. It can take values from this list: "h", "c1", "c2", "c3", "c4", "c5", "c6","c7". "h" and "c1" are human specific. If "all", all available gene set collections are loaded. If "none", MSigDB collections are excluded.
gsdb.gsets	character, a vector determines which gene set collections are loaded from the GeneSetDB. It takes "none", "all", "gsdbdis", "gsdbgo", "gsdbdrug", "gsdbpath" or "gsdbreg". "none" excludes the GeneSetDB collections. "all" includes all the GeneSetDB collections. "gsdbdis" to load the disease collection, "gsdbgo" to load the GO terms collection, "gsdbdrug" to load the drug/chemical collection, "gsdbpath" to load the pathways collection and "gsdbreg" to load the gene regulation collection.
go.part	logical, whether to partition the GO term collections into the three GO domains: CC, MF and BP or use the entire collection all together.
kegg.updated	logical, set to TRUE if you want to download the most recent KEGG pathways.
kegg.exclude	character, vector used to exclude KEGG pathways of specific type(s): Disease, Metabolism, Signaling. If "all", none fo the KEGG collections is included.
min.size	integer, the minium number of genes required in a testing gene set
updated	logical, set to TRUE if you want to download the most recent KEGG pathways.
exclude	character, vector used to exclude KEGG pathways of specific category. Accepted values are "Disease", "Metabolism", or "Signaling".
geneSets	character, a vector determines which gene set collections should be used. For MSigDB, it can take values from this list: "all", "h", "c1", "c2", "c3", "c4", "c5", "c6","c7". "c1" is human specific. For GeneSetDB, it takes "all", "gsdbdis", "gsdbgo", "gsdbdrug", "gsdbpath" or "gsdbreg". "gsdbdis" is to load the disease collection, "gsdbgo" to load the GO terms collection, "gsdbdrug" to load the drug/chemical collection, "gsdbpath" to load the pathways collection and "gsdbreg" to load the gene regulation collection. If "all", all available gene set collections are loaded.

geneIDs	character, a vector that stores the Gene IDs tagged in your dataset. The order of the Gene IDs must match those of the count/expression matrix row names. Gene IDs can be in any annotation, e.g., Symbols, Ensembl, etc., as soon as the parameter gsets uses the same Gene ID annotation.
gsets	list, list of gene sets. Each gene set is character vector of Enterz IDs. The names of the list should match the GeneSet column in the anno argument (if it is provided).
anno	list, dataframe that stores a detailed annotation for each gene set. Some of its fields can be ID, GeneSet, PubMed, URLs, etc. The GeneSet field is mandatory and should have the same names as the gsets' names.
label	character,a unique id that identifies the collection of gene sets
name	character,the collection name to be used in the EGSEA report
gmt.file	character, the path and name of the GMT file
anno.cols	integer, number of columns in the GMT file that are used for annotation. These columns should be inserted immediately after the second column.
anno.col.names	character, vector of the names of the annotation columns.

Details

buildIdx indexes the MSigDB, KEGG and GeneSetDB gene set collections, and loads gene set annotation.

buildKEGGIdx indexes the KEGG pathway gene sets and loads gene set annotation.

buildMSigDBIdx indexes the MSigDB gene sets and loads gene set annotation.

buildGeneSetDBIdx indexes the GeneSetDB gene sets and loads gene set annotation.

buildCustomIdx indexes newly created gene sets and attach gene set annotation if provided.

buildGMTIdx indexes newly created gene sets and attach gene set annotation if provided.

Value

buildIdx returns a list of gene set collection indexes, where each element of the list is an object of the class GSCollectionIndex.

buildKEGGIdx returns an object of the class GSCollectionIndex.

buildMSigDBIdx returns a list of gene set collection indexes, where each element of the list is an object of the class GSCollectionIndex.

buildGeneSetDBIdx returns a list of gene set collection indexes, where each element of the list is an object of the class GSCollectionIndex.

buildCustomIdx returns an object of the class GSCollectionIndex.

buildGMTIdx returns an object of the class GSCollectionIndex.

Examples

```
# example of buildIdx
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
         msigdb.gsets = c("h", "c5"),
   go.part = TRUE,
         kegg.exclude = c("Metabolism"))
names(gs.annots)
# example of buildKEGGIdx
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildKEGGIdx(entrezIDs=rownames(v$E), species="human")

# example of buildMSigDBIdx
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildMSigDBIdx(entrezIDs=rownames(v$E), species="human",
geneSets=c("h", "c2"))
names(gs.annots)
# example of buildGeneSetDBIdx
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildGeneSetDBIdx(entrezIDs=rownames(v$E), species="human")
names(gs.annots)


# example of buildCustomIdx
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
data(kegg.pathways)
gsets = kegg.pathways$human$kg.sets[1:50]
gs.annot = buildCustomIdx(geneIDs=rownames(v$E), gsets= gsets,
species="human")
class(gs.annot)


# example of buildGMTIdx
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
#gs.annot = buildGMTIdx(geneIDs=rownames(v$E), gsets= gmt.file,
#species="human")
#class(gs.annot)
```

egsea *Core functions to perform ensemble of gene set enrichment analysis (EGSEA)*

Description

egsea is the main function to carry out gene set enrichment analysis using the EGSEA algorithm. This function is aimed to extend the limma-voom pipeline of RNA-seq analysis.

egsea.cnt is the main function to carry out gene set enrichment analysis using the EGSEA algorithm. This function is aimed to use raw RNASeq count matrix to perform the EGSEA analysis.

egsea.ora is the main function to carry out over-representation analysis (ORA) on gene set collections using a list of genes.

egsea.ma is the main function to carry out gene set enrichment analysis using the EGSEA algorithm. This function is aimed to use a microarray expression matrix to perform the EGSEA analysis.

Usage

```
egsea(
  voom.results,
  contrasts = NULL,
  logFC = NULL,
  gs.annots,
  symbolsMap = NULL,
  baseGSEAs = egsea.base(),
  minSize = 2,
  display.top = 20,
  combineMethod = "wilkinson",
  combineWeights = NULL,
  sort.by = "p.adj",
  report.dir = NULL,
  kegg.dir = NULL,
  logFC.cutoff = 0,
  fdr.cutoff = 0.05,
  sum.plot.axis = "p.adj",
  sum.plot.cutoff = NULL,
  vote.bin.width = 5,
  num.threads = 4,
  report = TRUE,
  interactive = FALSE,
  keep.base = FALSE,
  verbose = FALSE,
  keep.limma = TRUE,
  keep.set.scores = FALSE
)

egsea.cnt(
```

```
  counts,
  group,
  design = NULL,
  contrasts = NULL,
  logFC = NULL,
  gs.annots,
  symbolsMap = NULL,
  baseGSEAs = egsea.base(),
  minSize = 2,
  display.top = 20,
  combineMethod = "wilkinson",
  combineWeights = NULL,
  sort.by = "p.adj",
  report.dir = NULL,
  kegg.dir = NULL,
  logFC.cutoff = 0,
  fdr.cutoff = 0.05,
  sum.plot.axis = "p.adj",
  sum.plot.cutoff = NULL,
  vote.bin.width = 5,
  num.threads = 4,
  report = TRUE,
  interactive = FALSE,
  keep.base = FALSE,
  verbose = FALSE,
  keep.limma = TRUE,
  keep.set.scores = FALSE
)

egsea.ora(
  geneIDs,
  universe = NULL,
  logFC = NULL,
  title = NULL,
  gs.annots,
  symbolsMap = NULL,
  minSize = 2,
  display.top = 20,
  sort.by = "p.adj",
  report.dir = NULL,
  kegg.dir = NULL,
  sum.plot.axis = "p.adj",
  sum.plot.cutoff = NULL,
  num.threads = 4,
  report = TRUE,
  interactive = FALSE,
  verbose = FALSE
)
```

```
egsea.ma(
  expr,
  group,
  probe.annot,
  probeMap.method = "avg",
  design = NULL,
  contrasts = NULL,
  logFC = NULL,
  gs.annots,
  baseGSEAs = egsea.base(),
  minSize = 2,
  display.top = 20,
  combineMethod = "wilkinson",
  combineWeights = NULL,
  sort.by = "p.adj",
  report.dir = NULL,
  kegg.dir = NULL,
  logFC.cutoff = 0,
  fdr.cutoff = 0.05,
  sum.plot.axis = "p.adj",
  sum.plot.cutoff = NULL,
  vote.bin.width = 5,
  num.threads = 4,
  report = TRUE,
  interactive = FALSE,
  keep.base = FALSE,
  verbose = FALSE,
  keep.limma = TRUE,
  keep.set.scores = FALSE
)
```

Arguments

`voom.results`	list, an EList object generated using the `voom` function, which it has at least three elements: `E` log2 normalized expression values, `design` design matrix, `targets` sample information, which must have a column named **group**. If a matrix `weights` is provided, it will be used in `limma`-based methods. Entrez Gene IDs should be used as row names.
`contrasts`	double, an N x L matrix indicates the contrasts of the linear model coefficients for which the test is required if the design matrix does not have an intercept. N is number of columns of the design matrix and L is number of contrasts. This matrix should be based on the primary factor of the design matrix. If the design matrix includes an intercept, this parameter can be a vector of integers that specify the columns of the design matrix. If this parameter is `NULL`, all pairwise comparisons based on `group` or `voom.results$targets$group` are created, assuming that `group` is the primary factor in the `design` matrix. Likewise, all the coefficients of the primary factor are used if the `design` matrix has an intercept.

logFC	double, an K x L matrix indicates the log2 fold change of each gene for each contrast. K is the number of genes included in the analysis. If logFC=NULL, the logFC values are estimated using the ebayes for each contrast. For egsea.ora, it can be a matrix or vector of the same length of entrezIDs. If logFC=NULL, 1 is used as a default value. Then, the regulation direction in heatmaps and pathway maps is not indicative of the gene regulation direction.
gs.annots	list, list of objects of class GSCollectionIndex. It is generated using one of these functions: buildIdx, buildMSigDBIdx, buildKEGGIdx, buildGeneSetDBIdx, and buildCustomIdx.
symbolsMap	dataframe, an K x 2 matrix stores the gene symbol of each Entrez Gene ID. The first column must be the Entrez Gene IDs and the second column must be the Gene Symbols. It is used for the heatmap visualization. In egsea and egsea.cnt, the number of rows should match that of the **voom.results** and **counts**, respectively. Default symbolsMap=NULL.
baseGSEAs	character, a vector of the gene set tests that should be included in the ensemble. Type egsea.base to see the supported GSE methods. By default, all supported methods are used.
minSize	integer, the minimum size of a gene set to be included in the analysis. Default minSize= 2.
display.top	integer, the number of top gene sets to be displayed in the EGSEA report. You can always access the list of all tested gene sets using the returned gsa list. Default is 20.
combineMethod	character, determines how to combine p-values from different GSEA method. Type egsea.combine() to see supported methods.
combineWeights	double, a vector determines how different GSEA methods will be weighted. Its values should range between 0 and 1. This option is not supported currently.
sort.by	character, determines how to order the analysis results in the stats table. Type egsea.sort() to see all available options. For egsea.ora, it takes "p.value", "p.adj" or "Significance".
report.dir	character, directory into which the analysis results are written out.
kegg.dir	character, the directory of KEGG pathway data file (.xml) and image file (.png). Default kegg.dir=paste0(report.dir, "/kegg-dir/").
logFC.cutoff	numeric, cut-off threshold of logFC and is used for the calculation of Sginificance Score and Regulation Direction. Default logFC.cutoff=0.
fdr.cutoff	numeric, cut-off threshold of DE genes and is used for the calculation of Significance Score and Regulation Direction. Default fdr.cutoff = 0.05.
sum.plot.axis	character, the x-axis of the summary plot. All the values accepted by the **sort.by** parameter can be used. Default sum.plot.axis="p.value".
sum.plot.cutoff	numeric, cut-off threshold to filter the gene sets of the summary plots based on the values of the **sum.plot.axis**. Default sum.plot.cutoff=NULL.
vote.bin.width	numeric, the bin width of the vote ranking. Default vote.bin.width=5.
num.threads	numeric, number of CPU cores to be used. Default num.threads=2.

`report` logical, whether to generate the EGSEA interactive report. It takes longer time to run. Default is True.

`interactive` logical, whether to generate interactive tables and plots. Note this might dramatically increase the size of the EGSEA report.

`keep.base` logical, whether to write out the results of the individual GSE methods. Default FALSE.

`verbose` logical, whether to print out progress messages and warnings.

`keep.limma` logical, whether to store the results of the limma analysis in the EGSEAResults object.

`keep.set.scores` logical, whether to calculate the gene set enrichment scores per sample for the methods that support this option, i.e., "ssgsea".

`counts` double, an K x M numeric matrix of read counts where genes are the rows and samples are the columns. In this case, TMM normalization is used to calculate the normalization factors. `counts` can be also a `DGEList` object. In this case, the normalization factors can be calculated by the user prior to invoking this method. This is REQUIRED.

`group` character, vector or factor giving the experimental group/condition for each sample/library. This is REQUIRED.

`design` double, an M x N numeric matrix giving the design matrix of the linear model fitting. N is the number of coefficients in model. If this parameter is `NULL`, `model.matrix(~0+group)` is used to create a deisgn matrix.

`geneIDs` character, a vector of Gene IDs to be tested for ORA. They must be Entrez IDs if **EGSEAdata** collections are used.

`universe` character, a vector of Enterz IDs to be used as a background list. If universe=NULL, the background list is created from the **AnnotationDbi** package.

`title` character, a short description of the experimental contrast.

`expr` double, an K x M numeric matrix of intensities where genes are the rows and samples are the columns. In this case, it is assumed that the expression values are already normalized log2 intensities and filtered, i.e. rows corresponding to control and low-quality probes removed. Row names of `expr` must match the first column in `probe.annot`. This is REQUIRED.

`probe.annot` double, an K x n numeric matrix where rows are probes, the first column contains probe IDs, the second column contains Entrez Gene IDs, and the third column (optional) contains the Gene Symbols. Symbols are used for the heatmap visualization Additional annotation columns can be added for the probes, e.g., Chromosome, QualityScore, etc. These additional columns are not used. This is REQUIRED.

`probeMap.method` character, the method to be used in mapping the Probe IDs into Enterz Gene IDs. Accepted methods include: "avg", "med", "var", "sum" and "iqr". EGSEA selects the probe with the highest average, median, variance, sum or IQR of expression, respectively, as a representative for each expressed gene.

Details

EGSEA, an acronym for *Ensemble of Gene Set Enrichment Analyses*, utilizes the analysis results of eleven prominent GSE algorithms from the literature to calculate collective significance scores for gene sets. These methods include: **ora**, **globaltest**, **plage**, **safe**, **zscore**, **gage**, **ssgsea**, **roast**, **fry**, **padog**, **camera** and **gsva**. The ora, gage, camera and gsva methods depend on a competitive null hypothesis while the remaining seven methods are based on a self-contained hypothesis. Conveniently, the algorithm proposed here is not limited to these twelve GSE methods and new GSE tests can be easily integrated into the framework. This function takes the voom object and the contrast matrix as parameters. The results of EGSEA can be seen using the `topSets` function.

EGSEA report is an interactive HTML report that is generated if `report=TRUE` to enable a swift navigation through the results of an EGSEA analysis. The following pages are generated for each gene set collection and contrast/comparison:
1. Stats Table page shows the detailed statistics of the EGSEA analysis for the `display.top` gene sets. It shows the EGSEA scores, individual rankings and additional annotation for each gene set. Hyperlinks to the source of each gene set can be seen in this table when they are available. The "Direction" column shows the regulation direction of a gene set which is calculated based on the `logFC`, which is either calculated from the limma differential expression analysis or provided by the user. The `logFC.cutoff` and `fdr.cutoff` are applied for this calculation. The calculations of the EGSEA scores can be seen in the references section. The method `topSets` can be used to generate custom Stats Table.
2. Heatmaps page shows the heatmaps of the gene fold changes for the gene sets that are presented in the Stats Table page. Red indicates up-regulation while blue indicates down-regulation. Only genes that appear in the input expression/count matrix are visualized in the heat map. Gene names are coloured based on their statistical significance in the `limma` differential expression analysis. The "Interpret Results" link below each heat map allows the user to download the original heat map values along with additional statistics from `limma` DE analysis (if available) so that they can be used to perform further analysis in R, e.g., customizing the heat map visualization. Additional heat maps can be generated and customized using the method `plotHeatmap`.
3. Summary Plots page shows the methods ranking plot along with the summary plots of EGSEA analysis. The method plot uses multidimensional scaling (MDS) to visualize the ranking of individual methods on a given gene set collection. The summary plots are bubble plots that visualize the distribution of gene sets based on the EGSEA Significance Score and another EGSEA score (default, p-value). Two summary plots are generated: ranking and directional plots. Each gene set is reprersented with a bubble which is coloured based on the EGSEA ranking (in ranking plots) or gene set regulation direction (in directional plots) and sized based on the gene set cardinality (in ranking plots) or EGSEA Significance score (in directional plots). Since the EGSEA "Significance Score" is proportional to the p-value and the absolute fold changes, it could be useful to highlight gene sets that have high Significance scores. The blue labels on the summary plot indicate gene sets that do not appear in the top 10 list of gene sets based on the "sort.by" argument (black labels) yet they appear in the top 5 list of gene sets based on the EGSEA "Significance Score". If two contrasts are provided, the rank is calculated based on the "comparison" analysis results and the "Significance Score" is calculated as the mean. If `sort.by = NULL`, the slot `sort.by` of the `object` is used to order gene sets. The method `plotSummary` can be used to customize the Summary plots by changing the x-axis score and filtering bubbles based on the values of the x-axis. The method `plotMethods` can be used to generate Methods plots.
4. Pathways page shows the KEGG pathways for the gene sets that are presented in the Stats Table of a KEGG gene set collection. The gene fold changes are overlaid on the pathway maps and

coloured based on the gene regulation direction: blue for down-regulation and red for up-regulation. The method `plotPathway` can be used to generate additional pathway maps. Note that this page only appears if a KEGG gene set collection is used in the EGSEA analysis.
5. Go Graphs page shows the Gene Ontology graphs for top 5 GO terms in each of three GO categories: Biological Processes (BP), Molecular Functions (MF), and Cellular Components (CC). Nodes are coloured based on the default `sort.by` score where red indicates high significance and yellow indicates low significance. The method `plotGOGraph` can be used to customize GO graphs by changing the default sorting score and the number of significance nodes that can be visualized. It is recommended that a small number of nodes is selected. Note that this page only appears if a Gene Ontology gene set collection is used, i.e., for the c5 collection from MSigDB or the gsdbgo collection from GeneSetDB.

Finally, the "Interpret Results" hyperlink in the EGSEA report allows the user to download the fold changes and limma analysis results and thus improve the interpretation of the results.
Note that the running time of this function significantly increseas when `report = TRUE`. For example, the analysis in the example section below was conducted on the 203 signaling and disease KEGG pathways using a MacBook Pro machine that had a 2.8 GHz Intel Core i7 CPU and 16 GB of RAM. The execution time varied between 23.1 seconds (single thread) to 7.9 seconds (16 threads) when the HTML report generation was disabled. The execution time took 145.5 seconds when the report generation was enabled using 16 threads.

`egsea.ora` takes a list of gene IDs and uses the gene set collections from **EGSEAdata** or a custom-built collection to find over-represented gene sets in this list. It takes the advantage of the existing EGSEA reporting capabilities and generate an interative report for the ORA analysis. The results can be explored using the `topSets` function.

Value

`egsea` returns an object of the class EGSEAResults, which stores the top gene sets and the detailed analysis results for each contrast and the comparative analysis results.

`egsea.cnt` returns an object of the class EGSEAResults, which stores the top gene sets and the detailed analysis results for each contrast and the comparative analysis results.

`egsea.ora` returns an object of the class EGSEAResults, which stores the top gene sets and the detailed analysis results.

`egsea.ma` returns an object of the class EGSEAResults, which stores the top gene sets and the detailed analysis results for each contrast and the comparative analysis results.

References

Monther Alhamdoosh, Milica Ng, Nicholas J. Wilson, Julie M. Sheridan, Huy Huynh, Michael J. Wilson, Matthew E. Ritchie; Combining multiple tools outperforms individual methods in gene set enrichment analyses. Bioinformatics 2017; 33 (3): 414-424. doi: 10.1093/bioinformatics/btw623

See Also

`topSets`, `egsea.base`, `egsea.sort`, `buildIdx`, `buildMSigDBIdx`, `buildKEGGIdx`, `buildGeneSetDBIdx`, and `buildCustomIdx`

Examples

```
# Example of egsea
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
contrasts = il13.data$contra
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
# set report = TRUE to generate the EGSEA interactive report
gsa = egsea(voom.results=v, contrasts=contrasts,  gs.annots=gs.annots,
         symbolsMap=v$genes, baseGSEAs=egsea.base()[-c(2,5,6,9,12)],
   display.top = 5, sort.by="avg.rank",
   report.dir="./il13-egsea-report",
         num.threads = 2, report = FALSE)
topSets(gsa)

# Example of egsea.cnt
library(EGSEAdata)
data(il13.data.cnt)
cnt = il13.data.cnt$counts
group = il13.data.cnt$group
design = il13.data.cnt$design
contrasts = il13.data.cnt$contra
genes = il13.data.cnt$genes
gs.annots = buildIdx(entrezIDs=rownames(cnt), species="human",
msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
# set report = TRUE to generate the EGSEA interactive report
gsa = egsea.cnt(counts=cnt, group=group, design=design, contrasts=contrasts,
         gs.annots=gs.annots,
         symbolsMap=genes, baseGSEAs=egsea.base()[-c(2,5,6,9,12)],
display.top = 5,
          sort.by="avg.rank",
report.dir="./il13-egsea-cnt-report",
         num.threads = 2, report = FALSE)
topSets(gsa)

# Example of egsea.ora
library(EGSEAdata)
data(il13.data)
voom.results = il13.data$voom
contrast = il13.data$contra
library(limma)
vfit = lmFit(voom.results, voom.results$design)
vfit = contrasts.fit(vfit, contrast)
vfit = eBayes(vfit)
top.Table = topTable(vfit, coef=1, number=Inf, p.value=0.05, lfc=1)
deGenes = as.character(top.Table$FeatureID)
logFC =  top.Table$logFC
names(logFC) = deGenes
gs.annots = buildIdx(entrezIDs=deGenes, species="human",
```

```
msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
# set report = TRUE to generate the EGSEA interactive report
gsa = egsea.ora(geneIDs=deGenes, universe=
as.character(voom.results$genes[,1]),
             logFC =logFC, title="X24IL13-X24",
gs.annots=gs.annots,
             symbolsMap=top.Table[, c(1,2)], display.top = 5,
              report.dir="./il13-egsea-ora-report", num.threads = 2,
    report = FALSE)
topSets(gsa)

# Example of egsea.ma
data(arraydata)
expr = arraydata$arrays$E
group = as.factor(arraydata$targets$Condition)
levels(group) = c("DPcreEzh2", "DPev", "LumcreEzh2", "Lumev")
probe.annot = arraydata$arrays$genes[-1]
design <- model.matrix(~0+ group + as.factor(arraydata$targets$Experiment))
colnames(design)[1:4] = levels(group)
colnames(design)[5:6] = c("Exp2", "Exp3")
contr = makeContrasts("DPEzh2KOvsWT" = DPcreEzh2-DPev,
                        "LumEzh2KOvsWT" = LumcreEzh2-Lumev,
                      levels=colnames(design))

gs.annots = buildIdx(entrezIDs=unique(probe.annot[, 2]),
    species="mouse",
   msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))

# set report = TRUE to generate the EGSEA interactive report
gsa = egsea.ma(expr=expr, group=group,
   probe.annot = probe.annot,
   design=design, contrasts=contr,
         gs.annots=gs.annots,
         baseGSEAs=egsea.base()[-c(2,5,6,9,12)],
   display.top = 5,
          sort.by="avg.rank",
   report.dir="./ezh2-egsea-ma-report",
         num.threads = 2, report = FALSE)
topSets(gsa)
```

`egsea.sort` *EGSEA auxiliary functions*

Description

It lists the accepted sorting methods for analysis results

It lists the p-value combining methods

It lists the supported GSEA methods. Since EGSEA base methods are implemented in the Bioconductor project, the most recent version of each individual method is always used.

This function writes out the official EGSEA package logo

Usage

```
egsea.sort()

egsea.combine()

egsea.base()

egsea.logo(out.dir = "./")
```

Arguments

out.dir	character, the target directory to which the logo will be written.

Details

These methods include: **ora**[1], **globaltest**[2], **plage**[3], **safe**[4], **zscore**[5], **gage**[6], **ssgsea**[7], **roast**[8], **fry**[8], **padog**[9], **camera**[10] and **gsva**[11]. The `ora`, `gage`, `camera` and `gsva` methods depend on a competitive null hypothesis while the remaining seven methods are based on a self-contained hypothesis. Conveniently, EGSEA is not limited to these twelve GSE methods and new GSE tests can be easily integrated into the framework.

Note: the execution time of base methods can vary depending on the size of gene set collections, number of samples, number of genes and number of contrasts. When a gene set collection of around 200 gene sets was tested on a dataset of 17,500 genes, 8 samples and 2 contrasts, the execution time of base methods in ascending order was as follows: `globaltest`; `safe`; `gage`; `gsva`; `zscore`; `plage`; `fry`; `camera`; `ora`; `ssgsea`; `padog`. When the same dataset was tested on a large gene set collection of 3,700 gene sets, the execution time of base methods in ascending order was as follows: `globaltest`; `camera`; `fry`; `zscore`; `plage`; `safe`; `gsva`; `ora`; `gage`; `padog`; `ssgsea`. Apparently, the size of gene set collection plays a key role in the execution time of most of the base methods. The reduction rate of execution time between the large and small gene set collections varied between 18% and 88%. `camera`, `fry`, `plage`, `zscore` and `ora` showed the least reduction rate of execution time. As a result, there is no guarantee that a single combination of base methods would run faster than other combinations. It is worth mentioning that our simulation results showed that the increasing number of base methods in the EGSEA analysis is desirable to achieve high performance.

This function generates a PNG file of the EGSEA logo, which can be used to acknowledge EGSEA in presentations/reports. The logo was designed by Roberto Bonelli from The Walter and Eliza Hall Institute of Medical Research.

Value

It returns a character vector of the accepted values for the sort.by argument in egsea

It returns a character vector of available methods for the combineMethod argument in egsea

It returns a character vector of supported GSE methods.

a PNG file.

References

[1] Tavazoie, S. et al. (1999). Systematic determination of genetic network architecture. Nature Genetics, 22(3), 281-5.
[2] Goeman, J. J. et al. (2004). A global test for groups of genes: testing association with a clinical outcome. Bioinformatics, 20(1), 93-9.
[3] Tomfohr, J. et al. (2005). Pathway level analysis of gene expression using singular value decomposition. BMC Bioinformatics, 6, 225.
[4] Barry, W. T. et al. (2005). Significance analysis of functional categories in gene expression studies: a structured permutation approach. Bioinformatics, 21(9), 1943-9.
[5] Lee, E. et al. (2008). Inferring pathway activity toward precise disease classification. PLoS Computational Biology, 4(11), e1000217.
[6] Luo, W. et al. (2009). GAGE: generally applicable gene set enrichment for pathway analysis. BMC Bioinformatics, 10, 161.
[7] Barbie, D. A. et al. (2009). Systematic RNA interference reveals that oncogenic KRASdriven cancers require TBK1. Nature, 462(7269), 108-12.
[8] Wu, D. et al. (2010). ROAST: rotation gene set tests for complex microarray experiments. Bioinformatics, 26(17), 2176-82.
[9] Tarca, A. L. et al. (2009). A novel signaling pathway impact analysis. Bioinformatics, 25(1), 75-82.
[10] Wu, D. and Smyth, G. K. (2012). Camera: a competitive gene set test accounting for inter-gene correlation. Nucleic Acids Research, 40(17), e133.
[11] Hanzelmann, S. et al. (2013). GSVA: gene set variation analysis for microarray and RNA-seq data. BMC Bioinformatics, 14, 7.

Examples

```
egsea.sort()

egsea.combine()

egsea.base()
```

`EGSEAResults` *The EGSEAResults class*

Description

The `EGSEAResults` class stores the results of an EGSEA analysis.

The opertator `$` extracts a slot from an object of class EGSEAResults.

`topSets` extracts a table of the top-ranked gene sets from an EGSEA analysis.

`show` displays the parameters of an EGSEAResults object

summary displays a brief summary of the analysis results stored in an EGSEAResults object

limmaTopTable returns a dataframe of the top table of the limma analysis for a given contrast.

generateReport creates an HTML report for the EGSEA analysis that enables users to seamlessly browse the test results.

getlimmaResults returns the linear model fit produced by limma::eBayes.

plotHeatmap generates a heatmap of fold changes for a selected gene set.

plotSummaryHeatmap generates a summary heatmap for the top n gene sets of the comparative analysis across multiple contrasts.

plotPathway generates a visual map for a selected KEGG pathway with the gene fold changes overalid on it.

plotMethods generates a multi-dimensional scaling (MDS) plot for the gene set rankings of different base GSE methods

plotSummary generates a Summary plot for EGSEA analysis.

plotGOGraph generates a graph of the top significant GO terms in a GO term collection, which could be c5 from MSigDB or Gene Ontolog from the GeneSetDB.

plotBars generates a multi-dimensional scaling (MDS) plot for the gene set rankings of different base GSE methods

showSetByname shows the details of a given gene set indicated by name.

showSetByID shows the details of a given gene set indicated by ID.

getSetScores returns a dataframe of the gene set enrichment scores per sample. This can be only calculated using specific base methods, namely, "ssgsea".

Usage

```
## S4 method for signature 'EGSEAResults'
x$name

topSets(
  object,
  gs.label = 1,
  contrast = 1,
  sort.by = NULL,
  number = 10,
  names.only = TRUE,
  verbose = TRUE
)

## S4 method for signature 'EGSEAResults'
show(object)

## S4 method for signature 'EGSEAResults'
summary(object)

limmaTopTable(object, contrast = 1)
```

```
generateReport(
  object,
  number = 20,
  sort.by = NULL,
  report.dir = NULL,
  kegg.dir = NULL,
  x.axis = NULL,
  x.cutoff = NULL,
  num.threads = 4,
  print.base = FALSE,
  interactive = FALSE,
  verbose = FALSE
)

getlimmaResults(object)

plotHeatmap(
  object,
  gene.set,
  gs.label = 1,
  contrast = 1,
  file.name = "heatmap",
  format = "pdf",
  fc.colors = c("#67A9CF", "#F7F7F7", "#EF8A62"),
  verbose = TRUE
)

plotSummaryHeatmap(
  object,
  gs.label = 1,
  number = 20,
  sort.by = NULL,
  hm.vals = NULL,
  show.vals = NULL,
  file.name = "sum_heatmap",
  format = "pdf",
  verbose = TRUE
)

plotPathway(
  object,
  gene.set,
  gs.label = 1,
  contrast = 1,
  file.name = "pathway",
  verbose = TRUE
)
```

```
plotMethods(
  object,
  gs.label = 1,
  contrast = 1,
  file.name = "methods.mds",
  format = "pdf",
  verbose = TRUE
)

plotSummary(
  object,
  gs.label = 1,
  contrast = 1,
  file.name = "summary",
  format = "pdf",
  x.axis = "p.adj",
  x.cutoff = NULL,
  sort.by = NULL,
  use.names = FALSE,
  interactive = FALSE,
  verbose = TRUE
)

plotGOGraph(
  object,
  gs.label = "c5",
  contrast = 1,
  sort.by = NULL,
  noSig = 5,
  file.name = "c5-top-",
  format = "pdf",
  verbose = TRUE
)

plotBars(
  object,
  gs.label = 1,
  contrast = 1,
  number = 20,
  sort.by = NULL,
  bar.vals = "p.adj",
  file.name = "bars_plot",
  format = "pdf",
  verbose = TRUE
)

showSetByName(object, gs.label = 1, set.name)
```

```
showSetByID(object, gs.label = 1, id)

getSetScores(object, gs.label = 1)
```

Arguments

x	EGSEAResults object, the analysis result object from egsea, egsea.cnt or egsea.ora.
name	character, the slot name
object	EGSEAResults object, the analysis result object from egsea, egsea.cnt or egsea.ora.
gs.label	the number or label of the gene set collection of interest.
contrast	contrast column number or column name specifying which contrast is of interest. if contrast = 0 or "comparison" and the number of contrasts greater than 1, the comparative gene sets are retruned.
sort.by	character, determines how to order the analysis results in the stats table. The accepted values depend on the function used to generate the EGSEA results.
number	integer, maximum number of gene sets to list
names.only	logical, whether to display the EGSEA statistics or not.
verbose	logical, whether to print out progress messages and warnings.
report.dir	character, directory into which the analysis results are written out.
kegg.dir	character, the directory of KEGG pathway data file (.xml) and image file (.png). Default kegg.dir=paste0(report.dir, "/kegg-dir/").
x.axis	character, the x-axis of the summary plot. All the values accepted by the **sort.by** parameter can be used. Default x.axis="p.value".
x.cutoff	numeric, cut-off threshold to filter the gene sets of the summary plots based on the values of the **x.axis**. Default x.cutoff=NULL.
num.threads	numeric, number of CPU cores to be used. Default num.threads=4.
print.base	logical, whether to write out the analysis results of the base methods. Default is False.
interactive	logical, whether to generate interactive tables and plots. Note this might dramatically increase the size of the EGSEA report.
gene.set	character, the name of the gene set. See the output of topSets.
file.name	character, the prefix of the output file name.
format	character, takes "pdf" or "png".
fc.colors	vector, determines the fold change colors of the heatmap. Three colors of the negative, zero and positive log fold changes, respectively, should be assigned. Default is c("#67A9CF", "#F7F7F7", "#EF8A62"). These colors were generated using `rev(RColorBrewer::brewer.pal(3, "RdBu"))`
hm.vals	character, determines which EGSEA score values are used to draw the map. Default is NULL which implies using the sort.by score.
show.vals	character, determines which EGSEA score values are displayed on the map. Default is NULL which does not show anything.

use.names	logical, determines whether to display the GeneSet IDs or GeneSet Names. Default is FALSE.
noSig	numeric, number of significant GO terms to be displayed. A number larger than 5 might not work due to the size of the generated graph.
bar.vals	character, determines which EGSEA score values are used to draw the bars. Default is NULL which implies using the sort.by score.
set.name	character, a vector of gene set names as they appear in topSets.
id	character, a vector of gene set IDs as they appears in the plotSummary.

Details

The EGSEAResults class is used by egsea, egsea.cnt and egsea.ora to store the results of an EGSEA analysis. This helps in mining the analysis results and generating customized tables and plots.

limmaTopTable output can be understood from limma::topTable.

EGSEA report is an interactive HTML report that is generated to enable a swift navigation through the results of an EGSEA analysis. The following pages are generated for each gene set collection and contrast/comparison:
1. Stats Table page shows the detailed statistics of the EGSEA analysis for the display.top gene sets. It shows the EGSEA scores, individual rankings and additional annotation for each gene set. Hyperlinks to the source of each gene set can be seen in this table when they are available. The "Direction" column shows the regulation direction of a gene set which is calculated based on the logFC, which is either calculated from the limma differential expression analysis or provided by the user. The method topSets can be used to generate custom Stats Table.
2. Heatmaps page shows the heatmaps of the gene fold changes for the gene sets that are presented in the Stats Table page. Red indicates up-regulation while blue indicates down-regulation. Only genes that appear in the input expression/count matrix are visualized in the heat map. Gene names are coloured based on their statistical significance in the limma differential expression analysis. The "Interpret Results" link below each heat map allows the user to download the original heat map values along with additional statistics from limma DE analysis (if available) so that they can be used to perform further analysis in R, e.g., customizing the heat map visualization. Additional heat maps can be generated and customized using the method plotHeatmap.
3. Summary Plots page shows the methods ranking plot along with the summary plots of EGSEA analysis. The method plot uses multidimensional scaling (MDS) to visualize the ranking of individual methods on a given gene set collection. The summary plots are bubble plots that visualize the distribution of gene sets based on the EGSEA Significance Score and another EGSEA score (default, p-value). Two summary plots are generated: ranking and directional plots. Each gene set is reprersented with a bubble which is coloured based on the EGSEA ranking (in ranking plots) or gene set regulation direction (in directional plots) and sized based on the gene set cardinality (in ranking plots) or EGSEA Significance score (in directional plots). Since the EGSEA "Significance Score" is proportional to the p-value and the absolute fold changes, it could be useful to highlight gene sets that have high Significance scores. The blue labels on the summary plot indicate gene sets that do not appear in the top 10 list of gene sets based on the "sort.by" argument (black labels) yet they appear in the top 5 list of gene sets based on the EGSEA "Significance Score". If two contrasts are provided, the rank is calculated based on the "comparison" analysis results and the "Significance Score" is calculated as the mean. The method plotSummary can be used to customize the Summary plots by changing the x-axis score and filtering bubbles based on the values of the

x-axis. The method plotMethods can be used to generate Method plots.
4. Pathways page shows the KEGG pathways for the gene sets that are presented in the Stats Table of a KEGG gene set collection. The gene fold changes are overlaid on the pathway maps and coloured based on the gene regulation direction: blue for down-regulation and red for up-regulation. The method plotPathway can be used to generate additional pathway maps. Note that this page only appears if a KEGG gene set collection is used in the EGSEA analysis.
5. Go Graphs page shows the Gene Ontology graphs for top 5 GO terms in each of three GO categories: Biological Processes (BP), Molecular Functions (MF), and Cellular Components (CC). Nodes are coloured based on the default sort.by score where red indicates high significance and yellow indicates low significance. The method plotGOGraph can be used to customize GO graphs by changing the default sorting score and the number of significance nodes that can be visualized. It is recommended that a small number of nodes is selected. Note that this page only appears if a Gene Ontology gene set collection is used, i.e., for the c5 collection from MSigDB or the gsdbgo collection from GeneSetDB.

Finally, the "Interpret Results" hyperlink in the EGSEA report allows the user to download the fold changes and limma analysis results and thus improve the interpretation of the results.

getlimmaResults's output can be manipulated using limma::topTable and limma::topTreat.

plotHeatmap fold changes are colored based on the fc.colors and only genes that appear in the EGSEA analysis are visualized in the heatmap. Gene names are coloured based on the statistical significance level from limma DE analysis.

plotSummaryHeatmap creates a summary heatmap for the rankings of top number gene sets of the comparative analysis across all the contrasts. The show.vals score can be displayed on the heatmap for each gene set. This can help to identify gene sets that are highly ranked/sgnificant across multiple contrasts.

plotSummary generates a Summmary Plot for an EGSEA analysis. Since the EGSEA "Significance Score" is proportional to the p-value and the absolute fold changes, it could be useful to highlight gene sets that have high Significance scores. The blue labels on the summary plot indicate gene sets that do not apear in the top 10 list of gene sets based on the "sort.by" argument (black labels) yet they appear in the top 5 list of gene sets based on the EGSEA "Significance Score". If two contrasts are provided, the rank is calculated based on the "comparison" analysis results and the "Significance Score" is calculated as the mean. If sort.by = NULL, the slot sort.by of the object is used to order gene sets.

Value

$ returns the selected slot.

topSets returns a dataframe of top gene sets with the calculated statistics for each if names.only = FALSE.

show does not return data.

summary does not return data.

limmaTopTable returns a dataframe.

generateReport does not return data but creates an HTML report.

getlimmaResults returns an MArrayLM object.

plotHeatmap does not return data but creates image and CSV files.

plotSummaryHeatmap does not return data but creates image and CSV files.

plotPathway does not return data but creates a file.

plotMethods does not reutrn data but creates an image file.

plotSummary does not return data but creates an image file.

plotGOGraph does not return data but creates an image file.

plotBars does not reutrn data but creates an image file.

showSetByName does not return data

showSetByID does not return data.

getSetScores returnsa a dataframe where rows are gene sets and columns are samples.

Slots

results list, EGSEA analysis results

limmaResults MArrayLM, is a limma linear fit model

contr.names character, the contrasts defined in the analysis

contrast double, an N x L matrix indicates the contrasts of the linear model coefficients for which the test is required. N is the number of columns of the design matrix and L is number of contrasts. Can be also a vector of integers that specify the columns of the design matrix.

sampleSize numeric, number of samples

gs.annots list, the gene set collection annotation index

baseMethods character, vector of base GSE methods

baseInfo list, additional information on the base methods (e.g., version).

combineMethod character, the p-value combining method

sort.by character, the results ordering argument

symbolsMap data.frame, the mapping between Entrez IDs and Gene Symbols

logFC matrix, the logFC matrix of contrasts

logFC.calculated character, indicates whether the logFC was calculated using limma DE analysis.

sum.plot.axis character, the x-axis of the summary plot

sum.plot.cutoff numeric, the cut-off threshold for the summary plot x-axis

report logical, whether the report was generated

report.dir character, the directory of the EGSEA HTML report

egsea.version character, the version of EGSEA package

egseaData.version character, the version of EGSEAdata package

Examples

```
# Exampple of EGSEAResults
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
print(gsa$baseMethods)

# Example of topSets
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
topSets(gsa,  gs.label="kegg",contrast=1, number = 10)
topSets(gsa,  gs.label=1, contrast=1, sort.by="ora", number = 10,
names.only=FALSE)
topSets(gsa,  gs.label="kegg",contrast=0, number = 10)

# Example of show
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
show(gsa)

# Example of summary
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
summary(gsa)

# Example of limmaTopTable
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
colnames(limmaTopTable(gsa))
head(limmaTopTable(gsa))

# Example of generateReport
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
# generateReport(gsa)

# Example of getlimmaResults
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
fit = getlimmaResults(gsa)
```

```
class(fit)
names(fit)

# Example of plotHeatmap
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
plotHeatmap(gsa, "Asthma", gs.label="kegg")
plotHeatmap(gsa, "Asthma", gs.label="kegg", contrast = "comparison",
file.name = "asthma.hm.cmp")

# Example of plotSummaryHeatmap
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
plotSummaryHeatmap(gsa, gs.label="kegg")

# Example of plotPathway
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
plotPathway(gsa, gs.label="kegg", "Asthma")
plotPathway(gsa, gs.label="kegg", "Asthma", contrast="comparison",
file.name = "asthma.map.cmp")

# Example of plotMethods
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
plotMethods(gsa)

# Example of plotSummary
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
plotSummary(gsa)
plotSummary(gsa, contrast=c(1,2), file.name = "summary.cmp")

# Example of plotGOGraph
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
plotGOGraph(gsa, sort.by="avg.rank")

# Example of plotBars
library(EGSEAdata)
data(il13.gsa)
```

```
gsa = il13.gsa
class(gsa)
plotBars(gsa)

# Example of showSetByName
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
showSetByName(gsa, "kegg", "Asthma")

# Example of showSetByID
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
showSetByID(gsa, "kegg", "hsa04060")

# Example of getSetScores
library(EGSEAdata)
data(il13.gsa)
gsa = il13.gsa
class(gsa)
head(getSetScores(gsa, "kegg"))
```

`GSCollectionIndex` *The GSCollectionIndex class*

Description

The `GSCollectionIndex` class stores an indexed gene set collection.

The operator `$` extracts a slot from an object of class GSCollectionIndex.

`summary` displays a brief summary of a gene set collection

`show` displays the details of a gene set collection

`getSetByName` retrieves the details of a given gene set indicated by name

`getSetByID` retrieves the details of a given gene set indicated by ID

Usage

```
## S4 method for signature 'GSCollectionIndex'
x$name

## S4 method for signature 'GSCollectionIndex'
summary(object)

## S4 method for signature 'GSCollectionIndex'
```

```
show(object)

getSetByName(object, set.name)

getSetByID(object, id)
```

Arguments

x	GSCollectionIndex, the indexed gene set collection generated from buildIdx, buildMSigDBIdx, buildKEGGIdx, buildGeneSetDBIdx, and buildCustomIdx.
name	character, the slot name
object	GSCollectionIndex, the indexed gene set collection generated from buildIdx, buildMSigDBIdx, buildKEGGIdx, buildGeneSetDBIdx, and buildCustomIdx.
set.name	character, a vector of gene set names as they appear in topSets.
id	character, a vector of gene set IDs as they appears in the plotSummary.

Details

The GSCollectionIndex is used by buildIdx, buildCustomIdx, buildKEGGIdx, buildMSigDBIdx and buildGeneSetDBIdx.

Value

$ returns the selected slot data.

summary does not return data.

show does not return data.

getSetByName returns a list of annotation records

getSetByID returns a list of the annotation records.

Slots

original list, the original gene sets

idx list, the gene set indexes

anno data.frame, the annotations of the gene sets

featureIDs character, vector of the original Entrez IDs that are used in the indexing procedure

species character, the species name

name character, the name of the gene set collection

label character, a label to distnguish this collection

version character, the database version from which the collection was extracted

date character, the update/download date of the database from other collections

Examples

```
# Example of GSCollectionIndex
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
   msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
print(gs.annots[[1]]$name)

# Example of summary
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
   msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
summary(gs.annots[[1]])

# Example of show
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
   msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
show(gs.annots[[1]])

# Example of getSetByName
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
   msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
getSetByName(gs.annots[[1]], "Asthma")

# Example of getSetByID
library(EGSEAdata)
data(il13.data)
v = il13.data$voom
gs.annots = buildIdx(entrezIDs=rownames(v$E), species="human",
   msigdb.gsets="none",
         kegg.updated=FALSE, kegg.exclude = c("Metabolism"))
getSetByID(gs.annots[[1]], "hsa04060")
```

Index

∗ **datasets**
    arraydata, 3
$,EGSEAResults-method *(*EGSEAResults*)*, 18
$,GSCollectionIndex-method
        *(*GSCollectionIndex*)*, 28

arraydata, 3

buildCustomIdx, *11*, *14*, *29*
buildCustomIdx *(*buildIdx*)*, 3
buildCustomIdx,egsea-index *(*buildIdx*)*, 3
buildGeneSetDBIdx, *11*, *14*, *29*
buildGeneSetDBIdx *(*buildIdx*)*, 3
buildGeneSetDBIdx,egsea-index
        *(*buildIdx*)*, 3
buildGMTIdx *(*buildIdx*)*, 3
buildGMTIdx,egsea-index *(*buildIdx*)*, 3
buildIdx, 3, *11*, *14*, *29*
buildIdx,egsea-index *(*buildIdx*)*, 3
buildKEGGIdx, *11*, *14*, *29*
buildKEGGIdx *(*buildIdx*)*, 3
buildKEGGIdx,egsea-index *(*buildIdx*)*, 3
buildMSigDBIdx, *11*, *14*, *29*
buildMSigDBIdx *(*buildIdx*)*, 3
buildMSigDBIdx,egsea-index *(*buildIdx*)*, 3

ebayes, *11*
EGSEA *(*EGSEA-package*)*, 2
egsea, 8, *22*
egsea,egsea-main *(*egsea*)*, 8
EGSEA-package, 2
egsea.base, *11*, *14*
egsea.base *(*egsea.sort*)*, 16
egsea.base,egsea-aux *(*egsea.sort*)*, 16
egsea.cnt, *22*
egsea.cnt *(*egsea*)*, 8
egsea.cnt,egsea-main *(*egsea*)*, 8
egsea.combine, *11*
egsea.combine *(*egsea.sort*)*, 16
egsea.combine,egsea-aux *(*egsea.sort*)*, 16
egsea.logo *(*egsea.sort*)*, 16
egsea.logo,egsea-aux *(*egsea.sort*)*, 16
egsea.ma *(*egsea*)*, 8
egsea.ora, *22*
egsea.ora *(*egsea*)*, 8
egsea.ora,egsea-main *(*egsea*)*, 8
egsea.sort, *11*, *14*, 16
egsea.sort,egsea-aux *(*egsea.sort*)*, 16
EGSEAResults, 18
EGSEAResults-class *(*EGSEAResults*)*, 18

generateReport *(*EGSEAResults*)*, 18
generateReport,EGSEAResults-method
        *(*EGSEAResults*)*, 18
getlimmaResults *(*EGSEAResults*)*, 18
getlimmaResults,EGSEAResults-method
        *(*EGSEAResults*)*, 18
getSetByID *(*GSCollectionIndex*)*, 28
getSetByID,GSCollectionIndex-method
        *(*GSCollectionIndex*)*, 28
getSetByName *(*GSCollectionIndex*)*, 28
getSetByName,GSCollectionIndex-method
        *(*GSCollectionIndex*)*, 28
getSetScores *(*EGSEAResults*)*, 18
getSetScores,EGSEAResults-method
        *(*EGSEAResults*)*, 18
GSCollectionIndex, 28
GSCollectionIndex-class
        *(*GSCollectionIndex*)*, 28

limmaTopTable *(*EGSEAResults*)*, 18
limmaTopTable,EGSEAResults-method
        *(*EGSEAResults*)*, 18

plotBars *(*EGSEAResults*)*, 18
plotBars,EGSEAResults-method
        *(*EGSEAResults*)*, 18
plotGOGraph *(*EGSEAResults*)*, 18
plotGOGraph,EGSEAResults-method
        *(*EGSEAResults*)*, 18

plotHeatmap *(*EGSEAResults*)*, 18
plotHeatmap,EGSEAResults-method
*(*EGSEAResults*)*, 18
plotMethods *(*EGSEAResults*)*, 18
plotMethods,EGSEAResults-method
*(*EGSEAResults*)*, 18
plotPathway *(*EGSEAResults*)*, 18
plotPathway,EGSEAResults-method
*(*EGSEAResults*)*, 18
plotSummary, *23*, *29*
plotSummary *(*EGSEAResults*)*, 18
plotSummary,EGSEAResults-method
*(*EGSEAResults*)*, 18
plotSummaryHeatmap *(*EGSEAResults*)*, 18
plotSummaryHeatmap,EGSEAResults-method
*(*EGSEAResults*)*, 18

show,EGSEAResults-method
*(*EGSEAResults*)*, 18
show,GSCollectionIndex-method
*(*GSCollectionIndex*)*, 28
showSetByID *(*EGSEAResults*)*, 18
showSetByID,EGSEAResults-method
*(*EGSEAResults*)*, 18
showSetByName *(*EGSEAResults*)*, 18
showSetByName,EGSEAResults-method
*(*EGSEAResults*)*, 18
summary,EGSEAResults-method
*(*EGSEAResults*)*, 18
summary,GSCollectionIndex-method
*(*GSCollectionIndex*)*, 28

topSets, *13, 14*, *22, 23*, *29*
topSets *(*EGSEAResults*)*, 18
topSets,EGSEAResults-method
*(*EGSEAResults*)*, 18

voom, *10*