

Package ‘augere.gsea’

September 10, 2026

Version 0.99.1

Date 2026-04-11

Title Automatic Generation of Gene Set Enrichment Analyses

Description

Implements pipelines for generating gene set enrichment analysis reports in the augere framework. This includes various competitive and self-contained gene sets from a variety of Bioconductor packages. Each pipeline function generates a self-contained Rmarkdown report with all of the steps required to reproduce the gene set enrichment analysis.

License MIT + file LICENSE

Imports stats, limma, S4Vectors, IRanges, augere.core, augere.de

Suggests testthat, knitr, rmarkdown, BiocStyle, airway, goseq, fgsea, SummarizedExperiment, org.Hs.eg.db, GO.db

VignetteBuilder knitr

RoxygenNote 7.3.3

Encoding UTF-8

URL <https://github.com/augere-bioinfo/augere.gsea>

BugReports <https://github.com/augere-bioinfo/augere.gsea/issues>

biocViews WorkflowManagement, ReportWriting, GeneSetEnrichment

git_url <https://git.bioconductor.org/packages/augere.gsea>

git_branch devel

git_last_commit a7cb67d

git_last_commit_date 2026-05-14

Repository Bioconductor 3.24

Date/Publication 2026-09-10

Author Aaron Lun [cre, aut] (ORCID: <<https://orcid.org/0000-0002-3564-4813>>)

Maintainer Aaron Lun <infinite.monkeys.with.keyboards@gmail.com>

Contents

augere.gsea-package	2
reexports	2
runContrast	2
runPrecomputed	6

Index**11**

augere.gsea-package	<i>augere.gsea: Automatic Generation of Gene Set Enrichment Analyses</i>
---------------------	--

Description

Implements pipelines for generating gene set enrichment analysis reports in the augere framework. This includes various competitive and self-contained gene sets from a variety of Bioconductor packages. Each pipeline function generates a self-contained Rmarkdown report with all of the steps required to reproduce the gene set enrichment analysis.

Author(s)

Maintainer: Aaron Lun <infinite.monkeys.with.keyboards@gmail.com> ([ORCID](#))

See Also

Useful links:

- <https://github.com/augere-bioinfo/augere.gsea>
- Report bugs at <https://github.com/augere-bioinfo/augere.gsea/issues>

reexports	<i>Objects exported from other packages</i>
-----------	---

Description

These objects are imported from other packages. Follow the links below to see their documentation.

augere.core [readResult](#), [wrapInput](#)

runContrast	<i>Differential gene set analyses with contrasts</i>
-------------	--

Description

Analyze a RNA-seq dataset for differential expression across a collection of gene sets. This requires access to the original values, unlike [runPrecomputed](#).

Usage

```
runContrast(
  x,
  sets,
  groups,
  comparison,
  covariates = NULL,
  block = NULL,
  subset.factor = NULL,
  subset.levels = NULL,
  subset.groups = TRUE,
  design = NULL,
  contrast = NULL,
  dc.block = NULL,
  robust = TRUE,
  quality = TRUE,
  trend = FALSE,
  methods = c("fry", "camera"),
  assay = 1,
  annotation = NULL,
  metadata = NULL,
  output.dir = "contrast",
  author = NULL,
  dry.run = FALSE,
  save.results = TRUE,
  suppress.plots = FALSE
)
```

Arguments

x	A SummarizedExperiment object containing a count matrix where genes and samples are in rows and columns, respectively. Alternatively, the output of wrapInput that refers to a SummarizedExperiment.
sets	A list or CharacterList of character vectors. Each vector represents a gene set and contains the identifiers for genes in that set. Identifiers should be consistent with the row names of x. The list may also be named with the names of the sets.
groups	String specifying the colData(x) column containing the grouping factor of interest, see processSimpleDesignMatrix for more details. This may be NULL for experimental designs with no groups, e.g., covariates only. Ignored if design and contrasts are provided.
comparison	Character vector of length no greater than 2. For length 2, this specifies the groups to be compared, whereas for length 1, this specifies the covariate to test. See processSimpleComparisons for more details. Alternatively, a named character vector with no more than 2 unique names, see processSimpleComparisons for more details. Unlike runVoom , a list of vectors is not accepted.
covariates	Character vector specifying the colData(x) columns containing continuous covariates of interest, see processSimpleDesignMatrix for more details. Ignored if design and contrasts are provided.

block	Character vector specifying the <code>colData(x)</code> columns containing additional (uninteresting) blocking factors, see <code>processSimpleDesignMatrix</code> for more details. Ignored if design and contrasts are provided.
subset.factor	String specifying the <code>colData(x)</code> column containing the factor to use for subsetting.
subset.levels	Vector containing the levels of the <code>subset.factor</code> to be retained.
subset.groups	Boolean indicating whether to automatically subset the dataset to only those samples assigned to groups in comparisons. Setting this to TRUE sacrifices some residual degrees of freedom for greater robustness against variability in irrelevant groups. Ignored if design and contrasts are provided. Also ignored if covariates is provided, as all samples are informative for a continuous covariate.
design	Matrix, function, or formula specifying the experimental design, see <code>processCustomDesignMatrix</code> for details. If this and contrasts are specified, groups, block, covariates, comparisons and <code>subset.groups</code> are ignored.
contrast	String, function or vector specifying a custom contrast, see <code>?processCustomContrasts</code> for more details. Unlike <code>runVoom</code> , a list of contrasts is not accepted.
dc.block	String specifying the blocking factor to use in <code>duplicateCorrelation</code> . Typically used for uninteresting factors that cannot be used in <code>block</code> as they are confounded with the factors of interest. No additional blocking is performed if NULL.
robust	Boolean indicating whether robust empirical Bayes shrinkage should be used in <code>eBayes</code> . Setting this to TRUE sacrifices some precision for improved robustness against genes with extreme variances.
quality	Boolean indicating whether quality weighting should be performed. This reduces the influence of low-quality samples at the cost of more computational work.
trend	Boolean indicating whether variances should be shrunk towards a trend in <code>eBayes</code> . Usually unnecessary as the observation weights already account for the mean-variance relationship.
methods	Character vector specifying the methods to run. This should contain at least one of the following: • "mroast", which calls <code>mroast</code> from the limma package. This is a self-contained gene set test where the null hypothesis is that the genes in the set are not differentially expressed. • "fry", which calls <code>fry</code> from the limma package. This is also a self-contained gene set test that is a fast approximation of <code>mroast</code>. • "camera", which calls <code>camera</code> from the limma package. This is a competitive gene set test where the null hypothesis is that the genes in the set are not more DE than the genes outside the set. • "romer", which calls <code>romer</code> from the limma package. This is also a competitive test that takes a different approach to accounting for correlations between genes in the same set.
assay	String or integer specifying the assay of <code>x</code> containing the count matrix.
annotation	Character vector specifying the columns of <code>mcols(sets)</code> to store in each result DataFrames.
metadata	Named list of additional metadata to store alongside each result.

<code>output.dir</code>	String containing the path to an output directory in which to write the Rmarkdown file and save results.
<code>author</code>	Character vector containing the names of the authors. If NULL, defaults to the current user.
<code>dry.run</code>	Boolean indicating whether to perform a dry run. This generates the Rmarkdown report in <code>output.dir</code> but does not execute the analysis.
<code>save.results</code>	Boolean indicating whether the results should be saved to file.
<code>suppress.plots</code>	Boolean indicating whether plots should be suppressed. This can be set to TRUE for faster execution.

Details

Some of the methods involve randomization, so for full reproducibility, users should call `set.seed` before running `runContrast`.

Note that, even if the user does not call `set.seed`, `runContrast` will automatically insert `set.seed` statements into the Rmarkdown report prior to any GSEA functions that involve randomization. Each `set.seed` call has a hard-coded seed to ensure that future compilation of the generated report will give the same result. However, different calls to `runContrast` will use different (randomly selected) seeds to avoid systematic biases. Thus, if full reproducibility of `runContrast` is required, users should set the seed themselves before calling `runContrast`.

Value

A Rmarkdown report named `report.Rmd` is written inside `output.dir`. This contains all commands used to reproduce the analysis.

If `dry.run=FALSE`, a list of [DataFrames](#) is returned where each DataFrame contains the enrichment results for a method. Each row corresponds to a gene set in `sets` and is named accordingly. All DataFrames are guaranteed to have (at least) the following fields:

- `NumGenes`, the number of genes in the set (after removing genes that were not tested).
- `Direction`, the net direction of the change in expression within the set (either "up" or "down").
- `PValue`, the p-value for enrichment in each gene set.
- `FDR`, the Benjamini-Hochberg-adjusted p-value.

Additional fields may be present for specific methods.

If `save.results=TRUE`, the results are saved in a `results` subdirectory of `output.dir`.

If `dry.run=FALSE`, only the report is created, and NULL is returned.

Author(s)

Aaron Lun

Examples

```
x <- augere.de::loadExampleDataset()
all.genes <- rownames(x)

sets <- list(
  A = sample(all.genes, 92),
  B = sample(all.genes, 212),
  C = sample(all.genes, 12),
```

```

    D = sample(all.genes, 38),
    E = sample(all.genes, 55)
  )

  output.dir <- tempfile()
  res <- runContrast(
    x,
    sets,
    group="dex",
    comparison=c("trt", "untrt"),
    output=output.dir
  )

  res
  list.files(output.dir, recursive=TRUE)

```

runPrecomputed	<i>GSEA on precomputed statistics</i>
----------------	---------------------------------------

Description

Run gene set enrichment analyses on precomputed statistics, usually from differential expression analyses. This uses competitive gene set tests where the enrichment of “interesting” genes within the set must be greater than that outside of the set.

Usage

```

runPrecomputed(
  x,
  sets,
  signif.field,
  signif.threshold,
  rank.field,
  methods = c("hypergeometric", "goseq", "fgsea", "cameraPR"),
  alternative = c("mixed", "up", "down", "either"),
  rank.sqrt = FALSE,
  sign.field = NULL,
  goseq.bias = "AveExpr",
  goseq.args = list(),
  fgsea.leading.edge = FALSE,
  fgsea.args = list(),
  geneSetTest.args = list(),
  cameraPR.args = list(),
  metadata = NULL,
  annotation = NULL,
  author = NULL,
  output.dir = "precomputed",
  dry.run = FALSE,
  save.results = TRUE,
  suppress.plots = FALSE
)

```

Arguments

x	A data frame or DataFrame object containing various test statistics (columns) for genes (rows). Rows should be named with the same gene identifiers used in sets. Rows may contain NA values, which will be removed prior to analysis. Rows should <i>not</i> be filtered to only the significant genes, this will be handled by <code>signif.threshold</code>. Users should not pass in a table containing only the significant results. Alternatively, an object returned by wrapInput that refers to a data frame or DataFrame.
sets	A list or CharacterList of character vectors. Each vector represents a gene set and contains the identifiers for genes in that set. Identifiers should be consistent with the row names of x. The list may also be named with the names of the sets.
signif.field	String specifying the column of x to define significant genes, typically the adjusted p-value. Only used for <code>method="hypergeometric"</code> and <code>"goseq"</code> .
signif.threshold	Number specifying the upper threshold for significance to apply to the statistics from <code>signif.field</code> . All genes with lower statistics are considered to be significant. Only used for <code>method="hypergeometric"</code> and <code>"goseq"</code> .
rank.field	String specifying the column of x containing test statistics for ranking, e.g., t-statistics, Z-scores. This is generally expected to be signed such that the values with the largest magnitude are most significant and the sign represents some kind of directionality. It may be unsigned if <code>sign.field</code> is provided, in which case the signs are used to convert the test statistics back to signed values; or if <code>alternative="mixed"</code> , in which case the signs are ignored by each method. Only used for <code>method="geneSetTest"</code> , <code>"fgsea"</code> and <code>"cameraPR"</code> .
methods	Character vector specifying the gene set testing methods to use. This can be any number of the following: • <code>"hypergeometric"</code>, a hypergeometric test for enrichment of significant genes in each set. The set of significant genes is determined by <code>signif.field</code> and <code>signif.threshold</code>. This is equivalent to a one-sided Fisher's exact test. • <code>"goseq"</code>, which uses goseq from the goseq package. This is much like the hypergeometric test but accounting for gene-specific biases in detection. • <code>"geneSetTest"</code>, which uses geneSetTest from the limma package. This ranks genes according to some statistic (see <code>rank.field</code>) and tests for differences in the ranking of genes in and outside of the set. • <code>"fgsea"</code>, which uses fgsea from the fgsea package. This also tests for differences in ranking but is faster than <code>geneSetTest</code>. • <code>"cameraPR"</code>, which uses cameraPR from the limma package. This also tests for differences in ranking while accounting for variance inflation due to correlations between genes.
alternative	String specifying the alternative hypothesis. This should be one of: • <code>"mixed"</code> tests for enrichment of any significant genes in each set, regardless of their direction. • <code>"up"</code> tests for enrichment of up-regulated genes in each set. • <code>"down"</code> tests for enrichment of down-regulated genes in each set. • <code>"either"</code> tests for enrichment of either up- or down-regulated genes in each set. Specifically, each gene set is tested separately for enrichment of

up- and down-regulated genes, and the results are combined into a single p-value. This differs from "mixed", which does not consider the sign at all and only cares about enrichment of significant genes of any direction in each set.

rank.sqrt	Boolean indicating whether to compute the square root of the rank.field statistic before restoring the sign with sign.field. For example, the F-statistic will be converted to a t-statistic while the likelihood ratio will be converted to a Z-score. Only used for method="geneSetTest", "fgsea" and "cameraPR".
sign.field	String specifying the column of x containing a signed effect size, e.g., the log-fold change. This is used to restore the sign to unsigned test statistics in rank.field. It will also be used to define up- or down-regulated genes in method="hypergeometric" and "goseq" when alternative is not "mixed".
goseq.bias	String specifying the column of x containing the per-gene detection bias. Defaults to the gene abundance as larger counts provide greater power for detecting differential expression, but if this is not available, the exonic gene length can also be used. Only used for method="goseq".
goseq.args	Named list of additional arguments to pass to goseq .
fgsea.leading.edge	Boolean indicating whether the "leading edge" should be stored from fgsea .
fgsea.args	Named list of additional arguments to pass to fgsea .
geneSetTest.args	Named list of additional arguments to pass to geneSetTest .
cameraPR.args	Named list of additional arguments to pass to cameraPR .
metadata	Named list of additional metadata to store with each result.
annotation	Character vector specifying the columns of mcols (sets) to store in each result DataFrames.
author	Character vector containing the names of the authors. If NULL, defaults to the current user.
output.dir	String containing the path to an output directory in which to write the Rmarkdown file and save results.
dry.run	Logical scalar indicating whether a dry run should be performed, This generates the Rmarkdown report in output.dir but does not execute the analysis.
save.results	Boolean indicating whether the results should be saved to file.
suppress.plots	Boolean indicating whether to suppress the generation of plots. This can be set to FALSE for faster pipeline execution.

Details

Some of the methods involve randomization, so for full reproducibility, users should call [set.seed](#) before running runPrecomputed.

Note that, even if the user does not call [set.seed](#), runPrecomputed will automatically insert [set.seed](#) statements into the Rmarkdown report prior to any GSEA functions that involve randomization. Each [set.seed](#) call has a hard-coded seed to ensure that future compilation of the generated report will give the same result. However, different calls to runPrecomputed will use different (randomly selected) seeds to avoid systematic biases. Thus, if full reproducibility of runPrecomputed is required, users should set the seed themselves before calling runPrecomputed.

Value

A Rmarkdown report named `report.Rmd` is written inside `output.dir`. This contains all commands used to reproduce the analysis.

If `dry.run=FALSE`, a list of [DataFrames](#) is returned where each `DataFrame` contains the enrichment results for a method. Each row corresponds to a gene set in `sets` and is named accordingly. All `DataFrames` are guaranteed to have (at least) the following fields:

- `NumGenes`, the number of genes in the set (after removing genes that were not tested).
- `PValue`, the p-value for enrichment in each gene set.
- `FDR`, the Benjamini-Hochberg-adjusted p-value.
- (if `alternative="either"`) `Direction`, whether the set is more enriched for "up"- or "down"-regulated genes.

Specific methods will have additional fields:

- For "fgsea", `ES` contains the enrichment score and `NES` contains the normalized enrichment score. If `fgsea.leading.edge=TRUE`, `LeadingEdge` will contain a [CharacterList](#) with the names of genes in the "leading edge" for each gene set.
- For "goseq" and "hypergeometric", `NumSig` contains the number of significant genes in each set. (If `alternative` is "up" or "down", this is filtered for significant genes of the desired sign.)

If `save.results=TRUE`, the results are saved in a `results` subdirectory of `output.dir`.

If `dry.run=FALSE`, only the report is created, and `NULL` is returned.

Author(s)

Aaron Lun

Examples

```
all.genes <- sprintf("gene-%i", seq_len(1000))

library(S4Vectors)
tab <- DataFrame(
  row.names=all.genes,
  AveExpr=rnorm(length(all.genes)),
  t=rnorm(length(all.genes)),
  PValue=runif(length(all.genes)),
  FDR=runif(length(all.genes))
)

sets <- list(
  A=sample(all.genes, 20),
  B=sample(all.genes, 300),
  C=sample(all.genes, 10),
  D=sample(all.genes, 500)
)

output.dir <- tempfile()
results <- runPrecomputed(
  tab,
  sets,
  signif.field="FDR",
```

```
      signif.threshold=0.05,  
      rank.field="t",  
      output=output.dir  
    )  
    results$hypergeometric  
    results$fgsea  
  
    list.files(output.dir, recursive=TRUE)
```

Index

* **internal**

reexports, [2](#)

augere.gsea (augere.gsea-package), [2](#)

augere.gsea-package, [2](#)

camera, [4](#)

cameraPR, [7](#), [8](#)

CharacterList, [3](#), [7](#), [9](#)

colData, [3](#), [4](#)

DataFrame, [5](#), [7](#), [9](#)

duplicateCorrelation, [4](#)

eBayes, [4](#)

fgsea, [7](#), [8](#)

fry, [4](#)

geneSetTest, [7](#), [8](#)

goseq, [7](#), [8](#)

mcols, [4](#), [8](#)

mroast, [4](#)

processCustomContrasts, [4](#)

processCustomDesignMatrix, [4](#)

processSimpleComparisons, [3](#)

processSimpleDesignMatrix, [3](#), [4](#)

readResult, [2](#)

readResult (reexports), [2](#)

reexports, [2](#)

romer, [4](#)

runContrast, [2](#)

runPrecomputed, [2](#), [6](#)

runVoom, [3](#), [4](#)

set.seed, [5](#), [8](#)

SummarizedExperiment, [3](#)

wrapInput, [2](#), [3](#), [7](#)

wrapInput (reexports), [2](#)