

Package ‘exploreSE’

August 22, 2026

Title Explorer for Your Bulk RNA Seq Analyses

Version 0.99.5

Description Interactive exploration of a Summarized experiment and associated DE analyses. A shiny app to inspect and interact with pre-calculated differential expression and functional enrichment results, and to facilitate comparisons between multiple models. This is meant to be a tool for exploration and discussion, rather than analysis.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.6.0)

Imports BiocGenerics, clusterProfiler, colourpicker, DeeDeeExperiment, DESeq2, dplyr, DT, forcats, ggplot2, htmltools, magrittr, matrixStats, methods, msigdb, org.Hs.eg.db, org.Mm.eg.db, plotly, purrr, readr, rlang, S4Vectors, shiny, shinyWidgets, stats, stringr, SummarizedExperiment, tibble, tidyr, tidyselect

Suggests airway, BiocStyle, knitr, rmarkdown, testthat (>= 3.0.0)

URL <https://github.com/jaspitzer/exploreSE>

BugReports <https://github.com/jaspitzer/exploreSE/issues>

biocViews ShinyApps, RNASeq, Visualization, GeneExpression, Pathways, GO

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

git_url <https://git.bioconductor.org/packages/exploreSE>

git_branch devel

git_last_commit c0866f8

git_last_commit_date 2026-08-18

Repository Bioconductor 3.24

Date/Publication 2026-08-21

Author Jasper Spitzer [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-9696-2092>>)

Maintainer Jasper Spitzer <jasspitzer135@gmail.com>

Contents

exploreSE-package	2
baseline	3
baseline_down_go	3
baseline_up_go	4
cell_controlled	4
controlled_down_go	5
controlled_up_go	5
exploreSE	6
get.gos	6
get.gsea	7
%>%	8
Index	9

exploreSE-package	<i>exploreSE: Explorer for Your Bulk RNA Seq Analyses</i>
-------------------	---

Description

Interactive exploration of a Summarized experiment and associated DE analyses. A shiny app to inspect and interact with pre-calculated differential expression and functional enrichment results, and to facilitate comparisons between multiple models. This is meant to be a tool for exploration and discussion, rather than analysis.

Author(s)

Maintainer: Jasper Spitzer <jasspitzer135@gmail.com> ([ORCID](#))

Authors:

- Jasper Spitzer <jasspitzer135@gmail.com> ([ORCID](#))

See Also

Useful links:

- <https://github.com/jaspitzer/exploreSE>
- Report bugs at <https://github.com/jaspitzer/exploreSE/issues>

baseline

DE results for airway - baseline

Description

Example DE results for airway data set

Format

A DESeqResults

Details

This is a DESeqResults object, with the simplest model, using ~dex as the design. Data is from the airway package, DE results are generated with DESeq2.

Value

A DESeqResults

baseline_down_go

Example GO enrichments for the baseline model

Description

GO enrichments based on the baseline DE results

Format

A EnrichResults

Details

GO enrichments are generated with clusterProfiler, filtered on down-regulated genes only; genes that underwent DE testing were taken as the universe.

The code to create said object can be found in the folder /inst/scripts in the exploreSE package, the file is called creating_dataset_examples.R.

Value

A EnrichResults object

`baseline_up_go`*Example GO enrichments for the baseline model*

Description

GO enrichments based on the baseline DE results

Format

A `EnrichResults`

Details

GO enrichments are generated with `clusterProfiler`, filtered on up-regulated genes only; genes that underwent DE testing were taken as the universe.

The code to create said object can be found in the folder `/inst/scripts` in the `exploreSE` package, the file is called `creating_dataset_examples.R`.

Value

A `EnrichResults` object

`cell_controlled`*DE results for airway - cell-controlled*

Description

Example DE results for airway data set

Format

A `DESeqResults`

Details

This is a `DESeqResults` object, using `~ cell + dex` as the design. Data is from the `airway` package, DE results are generated with `DESeq2`, using the `dex untrt vs trt` contrast

Value

A `DESeqResults`

controlled_down_go	<i>Example GO enrichments for the cell_controlled model</i>
--------------------	---

Description

GO enrichments based on the cell_controlled DE results

Format

A EnrichResults

Details

GO enrichments are generated with clusterProfiler, filtered on down-regulated genes only; genes that underwent DE testing were taken as the universe.

The code to create said object can be found in the folder /inst/scripts in the exploreSE package, the file is called creating_dataset_examples.R.

Value

A EnrichResults object

controlled_up_go	<i>Example GO enrichments for the cell controlled model</i>
------------------	---

Description

GO enrichments based on the cell_controlled DE results

Format

A EnrichResults

Details

GO enrichments are generated with clusterProfiler, filtered on up-regulated genes only; genes that underwent DE testing were taken as the universe.

The code to create said object can be found in the folder /inst/scripts in the exploreSE package, the file is called creating_dataset_examples.R.

Value

A EnrichResults object

 exploreSE

exploreSE

Description

This runs the explorer app. You can add a path to a .rds file or a the relevant R object as an argument to load it on start-up. The app looks for DE results in the metadata of the `metadata(object)$de_results` and, if it is a `DeeDeeExperiment`, in the DEA slots of the object. Similarly, the enrichment results are looked for in the `metadata(object)$fe_results` or FEA slots (`DeeDeeExperiment`). As the `DeeDeeExperiment` structure does a lot of work to clean and standardize the inputs and outputs for both DE and enrichment results, using it is highly recommended.

Usage

```
exploreSE(file = NULL, object = NULL)
```

Arguments

<code>file</code>	a string leading to a file you want loaded as a default
<code>object</code>	a <code>summarizedExperiment</code> object (or <code>DeeDeeExperiment</code>) to load in

Value

an application

Examples

```
app <- exploreSE()
if(interactive()){
  shiny::runApp(app, port = 1234)
}
```

 get.gos

Calculate GO Enrichments for a given comparison

Description

A function to calculate GO term enrichments from a given `summarizedExperiment` or `DeeDeeExperiment`. By default, it calculates separate enrichments for up- downregulated genes, at the cutoffs of $|\log FCI| > 1$ and adjusted p- value < 0.05 . It utilises the `clusterProfiler::enrichGO()` function.

Usage

```
get.gos(NAME, obj, species = "hs", gene_type = "SYMBOL")
```

Arguments

NAME	character, name of the DE comparison
obj	DeeDeeExperiment or SummarizedExperiment, the object
species	character, the species hs/mm
gene_type	character, the type of gene identifier

Value

a DeeDeeExperiment or SummarizedExperiment

Examples

```
library(airway)
library(DeeDeeExperiment)
data(airway)
data("baseline_de", package = "exploreSE")

se_with_de <- addDEA(DeeDeeExperiment(airway), baseline)
se_with_go <- get.gos(obj = se_with_de,
                      NAME = "baseline",
                      gene_type = "ENSEMBL")
```

get.gsea

Calculate GSEA Enrichments for a given comparison

Description

Calculates the GSEA enrichments with the counts within a summarizedExperiment or DeeDeeExperiment and adds them in the respective slots. It follows the **original signal-to-noise ratio of the method**; other approaches for ranking of genes are available in other implementations, such as the `clusterProfiler::GSEA()` implementation which calls for logFC ranked genes.

Usage

```
get.gsea(
  NAME,
  obj,
  type = "HALLMARK",
  conditions,
  species = "hs",
  condition_var = "condition",
  gene_type = "SYMBOL"
)
```

Arguments

NAME	character, name of the DE comparison
obj	DeeDeeExperiment or SummarizedExperiment, the object
type	character, the type of gene set, HALLMARK or REACTOME

conditions	a vector of length 2, which levels of the condition variable are being compared,
species	character, supported values are "hs" and "mm"
condition_var	symbol or character, the variable name for this comparison
gene_type	character, the type of gene identifier

Value

a DeeDeeExperiment or SummarizedExperiment

Examples

```
library(airway)
library(DeeDeeExperiment)
data(airway)
data("baseline_de", package = "exploreSE")

se_with_de <- addDEA(DeeDeeExperiment(airway), baseline)
se_with_gsea <- get.gsea(NAME = "baseline",
                        obj = se_with_de,
                        type = "HALLMARK",
                        condition_var = "dex",
                        conditions = c("trt", "untrt"),
                        gene_type = "ENSEMBL")
```

%>%	<i>Pipe operator</i>
-----	----------------------

Description

See `magrittr::%>%` for details.

Usage

```
lhs %>% rhs
```

Arguments

lhs	A value or the magrittr placeholder.
rhs	A function call using the magrittr semantics.

Value

The result of calling `rhs(lhs)`.

Examples

```
TRUE
```

Index

- * **internal**
 - %>%, [8](#)
 - exploreSE-package, [2](#)
 - %>%, [8](#), [8](#)
- baseline, [3](#)
- baseline_down_go, [3](#)
- baseline_up_go, [4](#)
- cell_controlled, [4](#)
- clusterProfiler::enrichGO(), [6](#)
- clusterProfiler::GSEA(), [7](#)
- controlled_down_go, [5](#)
- controlled_up_go, [5](#)
- exploreSE, [6](#)
- exploreSE-package, [2](#)
- get.gos, [6](#)
- get.gsea, [7](#)