Package ‘geyser’

May 26, 2026

Title Gene Expression displaYer of SummarizedExperiment in R

Version 1.5.0

Description Lightweight Expression displaYer (plotter / viewer) of SummarizedExperiment object in R. This package provides a quick and easy Shiny-based GUI to empower a user to use a SummarizedExperiment object to view

License CC0

LazyData false

biocViews Software, ShinyApps, GUI, GeneExpression

Imports bslib (>= 0.6.0), BiocStyle, ComplexHeatmap, cowplot, dplyr, DT, ggbeeswarm, ggplot2, ggrepel, ggh4x, htmltools, magrittr, pals, RColorBrewer, rlang, R.utils, shiny, shinyjs, S4Vectors, SummarizedExperiment, tibble, tidyselect, tidyr, yaml

Suggests airway, knitr, DESeq2, rmarkdown, stringr, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

URL `https://github.com/davemcg/geyser`

BugReports `https://github.com/davemcg/geyser/issues`

RoxygenNote 7.3.2

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

git_branch devel

git_last_commit b00fc35

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-25

Author David McGaughey [aut, cre] (ORCID: <`https://orcid.org/0000-0002-9224-2888`>)

Maintainer David McGaughey <`mcgaughey@gmail.com`>

Contents

`.exp_plot` *exp_plot*

Description

draws the expression box plot

Usage

```
.exp_plot(input, rse, data_source_name)
```

Arguments

`input`	From ui.R
`rse`	The rse object
`data_source_name`	The name of the loaded dataset for the caption

Details

Makes the box plot for the geyser Shiny app

Value

Returns a list with the $plot slot holding ggplot object and $grouping_length contains the number of features to scale the plot

Author(s)

David McGaughey

Examples

```
load(system.file('extdata/tiny_rse.Rdata', package = 'geyser'))
input <- list()
input$feature_col <- "row names"
input$features <- c("TYRP1 (ENSG00000107165.12)","OPN1LW (ENSG00000102076.9)")
input$groupings <- c('disease')
input$slot <- 'counts'
input$expression_scale <- TRUE
input$color_by <- 'tissue'
input$color_palette <- 'polychrome'
input$show_points <- TRUE
geyser:::.exp_plot(input, tiny_rse, "tiny_rse.Rdata")$plot
```

`.hm_plot` *hm_plot*

Description

draws the expression heatmap

Usage

```
.hm_plot(input, rse, data_source_name)
```

Arguments

`input`	From ui.R
`rse`	rse object
`data_source_name`	The name of the loaded dataset for the caption

Details

Makes the heatmap for the geyser Shiny app

Value

Returns a ComplexHeatmap object

Author(s)

David McGaughey

Examples

```
load(system.file('extdata/tiny_rse.Rdata', package = 'geyser'))
input <- list()
input$feature_col <- 'row names'
input$features <- c("TYRP1 (ENSG00000107165.12)","OPN1LW (ENSG00000102076.9)")
input$groupings <- c('disease')
input$slot <- 'counts'
input$expression_scale <- TRUE
input$row_clust <- TRUE
```

```
input$col_clust <- TRUE
geyser:::.hm_plot(input, tiny_rse, "tiny_rse.Rdata")$plot
```

`apply_config` *Configuration Helper Functions for Geyser*

Description

Functions to handle loading, saving, and applying configurations

Apply a configuration to the Shiny session

Usage

```
apply_config(config, session, rv)
```

Arguments

`config`	List containing configuration parameters
`session`	Shiny session object
`rv`	reactiveValues object containing app state

`apply_embedded_config` *Apply embedded config from SummarizedExperiment if present*

Description

Apply embedded config from SummarizedExperiment if present

Usage

```
apply_embedded_config(rse, session, rv)
```

Arguments

`rse`	SummarizedExperiment object
`session`	Shiny session object
`rv`	reactiveValues object

`apply_pending_group_filters` *Apply pending group filters to dynamic UI*

Description

Apply pending group filters to dynamic UI

Usage

```
apply_pending_group_filters(rv, input, session)
```

Arguments

`rv`	reactiveValues object
`input`	Shiny input object
`session`	Shiny session object

`apply_pending_sample_filter` *Apply pending sample filter to table*

Description

Apply pending sample filter to table

Usage

```
apply_pending_sample_filter(rv, input)
```

Arguments

`rv`	reactiveValues object
`input`	Shiny input object

`create_config_download_handler` *Create configuration download handler*

Description

Create configuration download handler

Usage

```
create_config_download_handler(input)
```

Arguments

`input`	Shiny input object

`extract_config_from_metadata` *Extract geyser config from SummarizedExperiment metadata*

Description

Extract geyser config from SummarizedExperiment metadata

Usage

```
extract_config_from_metadata(rse)
```

Arguments

`rse`	SummarizedExperiment object

Value

List containing config or NULL if not found

`geyser` *geyser*

Description

Run shiny app to use SummarizedExperiment object to display genomics data

Usage

```
geyser(
  rse = NULL,
  app_name = "geyser",
  primary_color = "#3A5836",
  secondary_color = "#d5673e",
  computer_data_dir = NULL
)
```

Arguments

`rse`	SummarizedExperiment object. If NULL (the default), the app will start with a file upload screen.
`app_name`	Title name that goes on the top left of the Shiny app
`primary_color`	The title bar color
`secondary_color`	The plot action button color
`computer_data_dir`	Optional folder path to existing SummarizedExperiment RDS files.

Value

A Shiny app object

Examples

```
if (interactive()) {
  load(system.file('extdata/tiny_rse.Rdata', package = 'geyser'))
  geyser(tiny_rse, "Test RSE")
}
```

`setup_config_preset_observer`

Setup configuration preset observer

Description

Setup configuration preset observer

Usage

```
setup_config_preset_observer(input, session, rv)
```

Arguments

`input`	Shiny input object
`session`	Shiny session object
`rv`	reactiveValues object

`setup_config_upload_observer`

Setup configuration upload observer

Description

Setup configuration upload observer

Usage

```
setup_config_upload_observer(input, session, rv)
```

Arguments

`input`	Shiny input object
`session`	Shiny session object
`rv`	reactiveValues object

Index