

Package ‘chevreulShiny’

December 17, 2025

Type Package

Title Tools for managing SingleCellExperiment objects as projects

Version 1.3.0

Description Tools for managing SingleCellExperiment objects as projects.

Includes functions for analysis and visualization of single-cell data.

Also included is a shiny app for visualization of pre-processed scRNA data.

Supported by NIH grants R01CA137124 and R01EY026661 to David Cobrinik.

License MIT + file LICENSE

URL <https://github.com/whtns/chevreulShiny>,

<https://whtns.github.io/chevreulShiny/>

Date 2024-03-24

BugReports <https://github.com/cobriniklab/chevreulShiny/issues>

Depends R (>= 4.5.0), SingleCellExperiment, shiny (>= 1.6.0),
shinydashboard, chevreulProcess, chevreulPlot

Imports alabaster.base, clustree, ComplexHeatmap, DataEditR (>= 0.0.9), DBI, dplyr, DT, EnhancedVolcano, fs, future, ggplot2, ggplotify, grDevices, methods, patchwork, plotly, purrr, rappdirs, readr, RSQLite, S4Vectors, scales, shinyFiles, shinyhelper, shinyjs, shinyWidgets, stats, stringr, tibble, tidyr, tidyselect, utils, waiter, wiggleplotr

Suggests BiocStyle, knitr, RefManageR, rmarkdown, testthat (>= 3.0.0),
EnsDb.Mmusculus.v79, EnsDb.Hsapiens.v86

VignetteBuilder knitr

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

biocViews Coverage, RNASeq, Sequencing, Visualization, GeneExpression, Transcription, SingleCell, Transcriptomics, Normalization, Preprocessing, QualityControl, DimensionReduction, DataImport

Config/testthat/edition 3

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

git_branch devel

git_last_commit 7845161

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-12-17

Author Kevin Stachelek [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-2085-695X>>),
Bhavana Bhat [aut]

Maintainer Kevin Stachelek <kevin.stachelek@gmail.com>

Contents

chevreulShiny-package	3
append_to_project_db	3
cc.genes.cyclone	4
chevreulApp	5
create_project_db	5
create_proj_matrix	6
ensembl_version	6
get_transcripts_from_sce	7
grch38	7
grch38_tx2gene	8
human_to_mouse_homologs	9
list_plot_types	9
load_alabaster_from_proj	10
load_alabaster_path	10
load_bigwigs	11
make_bigwig_db	11
make_chevreul_clean_names	12
metadata_from_batch	12
minimalChevreulApp	13
plotly_settings	14
plot_gene_coverage_by_var	14
read_project_db	15
save_sce	16
small_example_dataset	16
subset_by_colData	17
tiny_sce	17
unite_metadata	18
update_project_db	18

Index

20

chevreulShiny-package *chevreulShiny: Tools for managing SingleCellExperiment objects as projects*

Description

Tools for managing SingleCellExperiment objects as projects. Includes functions for analysis and visualization of single-cell data. Also included is a shiny app for visualization of pre-processed scRNA data. Supported by NIH grants R01CA137124 and R01EY026661 to David Cobrinik.

Author(s)

Maintainer: Kevin Stachelek <kevin.stachelek@gmail.com> ([ORCID](#))

Authors:

- Bhavana Bhat <bhavanabhat29@gmail.com>

See Also

Useful links:

- <https://github.com/whtns/chevreulShiny>
- <https://whtns.github.io/chevreulShiny/>
- Report bugs at <https://github.com/cobriniklab/chevreulShiny/issues>

append_to_project_db *Update a database of chevreulShiny projects*

Description

Append projects to database

Usage

```
append_to_project_db(  
  new_project_path,  
  cache_location = "~/cache/chevreul",  
  sqlite_db = "single-cell-projects.db",  
  verbose = TRUE  
)
```

Arguments

new_project_path new project path

cache_location Path to cache "~/cache/chevreul"

sqlite_db sqlite db

verbose print messages

Value

a sqlite database with SingleCellExperiment objects

cc.genes.cyclone	<i>Cyclone cell cycle pairs by symbol</i>
------------------	---

Description

cell cycle genes with paired expression represented by HGNC symbol

Usage

cc.genes.cyclone

Format

a list of dataframes with G1, G2, and S gene expression

- G1** G1 gene symbols
- G2** G2 gene symbols
- S** S gene symbols ...

Source

cyclone

chevreulApp	Create a shiny app for a project on disk
-------------	--

Description

Create a shiny app for a project on disk

Usage

```
chevreulApp(  
  preset_project,  
  appTitle = "chevreul",  
  organism_type = "human",  
  futureMb = 13000,  
  db_name = "single-cell-projects.db"  
)
```

Arguments

preset_project	A preloaded project to start the app with
appTitle	A title of the App
organism_type	human or mouse
futureMb	amount of Mb allocated to future package
db_name	sqlite database with list of saved SingleCellExperiment objects

Value

a shiny app

create_project_db	Create a database of chevreulShiny projects
-------------------	---

Description

Create a database containing chevreulShiny projects

Usage

```
create_project_db(  
  cache_location = "~/cache/chevreul",  
  sqlite_db = "single-cell-projects.db",  
  verbose = TRUE  
)
```

Arguments

cache_location Path to cache "~/cache/chevreul"
sqlite_db Database to be created
verbose print messages

Value

a sqlite database with SingleCellExperiment objects

create_proj_matrix	<i>Create a Table of single Cell Projects</i>
--------------------	---

Description

Uses a list of projects to create a matrix of single cell projects

Usage

```
create_proj_matrix(proj_list)
```

Arguments

proj_list List of projects

Value

a tibble of single cell projects

ensembl_version	<i>Ensembl version used for build</i>
-----------------	---------------------------------------

Description

Ensembl version used for build

Usage

```
ensembl_version
```

Format

An object of class character of length 1.

Source

<http://www.ensembl.org/>

Examples

```
# ensembl_version
```

```
get_transcripts_from_sce
```

Get Transcripts in object

Description

Get transcript ids in objects for one or more gene of interest

Usage

```
get_transcripts_from_sce(object, gene)
```

Arguments

object	A SingleCellExperiment object
gene	Gene of interest

Value

transcripts constituting a gene of interest in a SingleCellExperiment object

```
grch38
```

Human annotation data

Description

Human (*Homo sapiens*) annotations based on genome assembly GRCH38 from Ensembl.

Usage

```
grch38
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 76062 rows and 9 columns.

Details

Variables:

- `ensgene`
- `entrez`
- `symbol`
- `chr`
- `start`
- `end`
- `strand`
- `biotype`
- `description`

Source

http://ensembl.org/homo_sapiens

Examples

```
data("grch38")
head(grch38)
```

grch38_tx2gene	<i>Human transcripts to genes</i>
----------------	-----------------------------------

Description

Lookup table for converting Human (*Homo sapiens*) Ensembl transcript IDs to gene IDs based on genome assembly GRCH38 from Ensembl.

Usage

```
grch38_tx2gene
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 277081 rows and 2 columns.

Details

Variables:

- `enstxp`
- `ensgene`

Source

http://ensembl.org/homo_sapiens

Examples

```
data(grch38_tx2gene)
head(grch38_tx2gene)
```

human_to_mouse_homologs

Gene Homologs Between Human and Mouse

Description

Homologs drawn from Biomart

Usage

```
human_to_mouse_homologs
```

Format

A data frame with 23188 rows and 2 columns

HGNC.symbol human gene symbols

MGI.symbol mouse gene symbols ...

Source

bioMart

list_plot_types

Collate list of variables to be plotted

Description

Collate list of variables to be plotted

Usage

```
list_plot_types(object)
```

Arguments

object a SingleCellExperiment object

Value

plot_types a list of category_vars or continuous_vars

Examples

```
data(small_example_dataset)
list_plot_types(small_example_dataset)
```

```
load_alabaster_from_proj
```

Load SingleCellExperiment Files from a single project path

Description

Load SingleCellExperiment Files from a single project path

Usage

```
load_alabaster_from_proj(proj_dir, ...)
```

Arguments

proj_dir	project directory
...	extra args passed to load_alabaster_path

Value

a SingleCellExperiment object

```
load_alabaster_path
```

Read in Gene and Transcript SingleCellExperiment Objects

Description

Read in Gene and Transcript SingleCellExperiment Objects

Usage

```
load_alabaster_path(proj_dir = getwd(), prefix = "unfiltered")
```

Arguments

proj_dir	path to project directory
prefix	default "unfiltered"

Value

a SingleCellExperiment object

load_bigwigs	<i>Load Bigwigs</i>
--------------	---------------------

Description

Load a tibble of bigwig file paths by cell id

Usage

```
load_bigwigs(object, bigwig_db = "~/cache/chevreul/bw-files.db")
```

Arguments

object	A object
bigwig_db	Sqlite database of bigwig files

Value

a vector of bigwigs file paths

make_bigwig_db	<i>Make Bigwig Database</i>
----------------	-----------------------------

Description

Make Bigwig Database

Usage

```
make_bigwig_db(  
  new_project = NULL,  
  cache_location = "~/cache/chevreul/",  
  sqlite_db = "bw-files.db"  
)
```

Arguments

new_project	Project directory
cache_location	Path to cache "~/cache/chevreul"
sqlite_db	sqlite db containing bw files

Value

a sqlite database of bigwig files for cells in a SingleCellExperiment object

`make_chevreul_clean_names`*Clean Vector of chevreulShiny Names*

Description

Cleans names of objects provided in a vector form

Usage

```
make_chevreul_clean_names(myvec)
```

Arguments

`myvec` A vector of object names

Value

a clean vector of object names

Examples

```
data(small_example_dataset)
make_chevreul_clean_names(colnames(
  get_colData(small_example_dataset)))
```

`metadata_from_batch`*Retrieve Metadata from Batch*

Description

Retrieve Metadata from Batch

Usage

```
metadata_from_batch(
  batch,
  projects_dir = "/dataVolume/storage/single_cell_projects",
  db_path = "single-cell-projects.db"
)
```

Arguments

`batch` batch
`projects_dir` path to project dir
`db_path` path to .db file

Value

a tibble with cell level metadata from a SingleCellExperiment object

minimalChevreulApp	Create a minimal chevreulShiny app using SingleCellExperiment input
--------------------	---

Description

Create a minimal chevreulShiny app using SingleCellExperiment input

Usage

```
minimalChevreulApp(  
  single_cell_sce = NULL,  
  appTitle = NULL,  
  organism_type = "human",  
  futureMb = 13000,  
  db_name = "single-cell-projects.db"  
)
```

Arguments

single_cell_sce	a singlecell object
appTitle	a title for the app
organism_type	human or mouse
futureMb	the megabytes available for the future package
db_name	a database of bigwig files

Value

a minimal chevreulShiny app

Examples

```
if (interactive() ) {  
  data("tiny_sce")  
  minimalChevreulApp(tiny_sce)  
}
```

plotly_settings	<i>Plotly settings</i>
-----------------	------------------------

Description

Change settings of a plotly plot

Usage

```
plotly_settings(plotly_plot, width = 600, height = 700)
```

Arguments

plotly_plot	A plotly plot
width	Default set to '600'
height	Default set to '700'

Value

a plotly plot with settings changed

plot_gene_coverage_by_var	<i>Plot BigWig Coverage for Genes of Interest by a Given Variable</i>
---------------------------	---

Description

Plot BigWig coverage for genes of interest colored by a given variable

Usage

```
plot_gene_coverage_by_var(
  genes_of_interest = "NRL",
  cell_metadata,
  bigwig_tbl,
  group_by = "batch",
  values_of_interest = NULL,
  organism = c("human", "mouse"),
  edb = NULL,
  heights = c(3, 1),
  scale_y = "log10",
  reverse_x = FALSE,
  start = NULL,
  end = NULL,
  summarize_transcripts = FALSE,
  ...
)
```

Arguments

genes_of_interest	Gene of interest
cell_metadata	a dataframe with cell metadata from object
bigwig_tbl	a tibble with colnames "name", "bigWig", and "sample_id" matching the file-name, absolute path, and sample name of each cell in the cell_metadata
group_by	Variable to color by
values_of_interest	values of interest
organism	human (default) or mouse
edb	ensembl db object
heights	The heights of each row in the grid of plot
scale_y	whether to scale coverage
reverse_x	whether to reverse x axis
start	start coordinates
end	end coordinates
summarize_transcripts	whether to summarize transcript counts
...	extra arguments passed to plotCoverageFromEnsemblDb

Value

a ggplot with coverage faceted by group_by

read_project_db	<i>Read a database of chevreulShiny projects</i>
-----------------	--

Description

Reads database of chevreulShiny projects to a data frame

Usage

```
read_project_db(
  cache_location = "~/cache/chevreul",
  sqlite_db = "single-cell-projects.db",
  verbose = TRUE
)
```

Arguments

cache_location	Path to cache "~/cache/chevreul"
sqlite_db	sqlite db
verbose	print messages

Value

a tibble with SingleCellExperiment objects

save_sce	Save object to /output/sce/_sce.rds
----------	-------------------------------------

Description

Save object to /output/sce/_sce.rds

Usage

```
save_sce(object, prefix = "unfiltered", proj_dir = getwd())
```

Arguments

- object a SingleCellExperiment object
- prefix a prefix for saving
- proj_dir path to a project directory

Value

a path to an rds file containing a SingleCellExperiment object

small_example_dataset	Small example SingleCellExperiment
-----------------------	------------------------------------

Description

created with scuttle::mockSCE

Usage

```
small_example_dataset
```

Format

An SCE with 200 cells and 1000 genes

Source

scuttle::mockSCE

subset_by_colData	<i>Subset by new colData</i>
-------------------	------------------------------

Description

Subset the object using new colData

Usage

```
subset_by_colData(colData_path, object)
```

Arguments

colData_path	Path to new colData
object	A object

Value

a SingleCellExperiment object

tiny_sce	<i>Tiny example SingleCellExperiment</i>
----------	--

Description

subset to only NRL from chevreuldata::human_gene_transcript_sce()

Usage

```
tiny_sce
```

Format

An SCE with only expression of NRL gene and NRL transcripts

Source

```
chevreuldata::human_gene_transcript_sce()
```

unite_metadata	<i>Unite metadata</i>
----------------	-----------------------

Description

Unite metadata

Usage

```
unite_metadata(object, group_bys)
```

Arguments

object	A SingleCellExperiment object
group_bys	A feature or variable to combine

Value

a SingleCellExperiment object with Idents formed from concatenation of group_bys

Examples

```
data(small_example_dataset)
unite_metadata(small_example_dataset, "Mutation_Status")
```

update_project_db	<i>Update a database of chevrelShiny projects</i>
-------------------	---

Description

Add new/update existing projects to the database by recursing fully

Usage

```
update_project_db(
  projects_dir = NULL,
  cache_location = "~/cache/chevreul",
  sqlite_db = "single-cell-projects.db",
  verbose = TRUE
)
```

Arguments

<code>projects_dir</code>	The project directory to be updated
<code>cache_location</code>	Path to cache "~/cache/chevreul"
<code>sqlite_db</code>	sqlite db
<code>verbose</code>	print messages

Value

a sqlite database with SingleCellExperiment objects

Index

- * **datasets**
 - cc.genes.cyclone, [4](#)
 - ensembl_version, [6](#)
 - grch38, [7](#)
 - grch38_tx2gene, [8](#)
 - human_to_mouse_homologs, [9](#)
 - small_example_dataset, [16](#)
 - tiny_sce, [17](#)
- * **internal**
 - chevreulShiny-package, [3](#)
- append_to_project_db, [3](#)
- cc.genes.cyclone, [4](#)
- chevreulApp, [5](#)
- chevreulShiny (chevreulShiny-package), [3](#)
- chevreulShiny-package, [3](#)
- create_proj_matrix, [6](#)
- create_project_db, [5](#)
- ensembl_version, [6](#)
- get_transcripts_from_sce, [7](#)
- grch38, [7](#)
- grch38_tx2gene, [8](#)
- human_to_mouse_homologs, [9](#)
- list_plot_types, [9](#)
- load_alabaster_from_proj, [10](#)
- load_alabaster_path, [10](#)
- load_bigwigs, [11](#)
- make_bigwig_db, [11](#)
- make_chevreul_clean_names, [12](#)
- metadata_from_batch, [12](#)
- minimalChevreulApp, [13](#)
- plot_gene_coverage_by_var, [14](#)
- plotly_settings, [14](#)
- read_project_db, [15](#)
- save_sce, [16](#)
- small_example_dataset, [16](#)
- subset_by_colData, [17](#)
- tiny_sce, [17](#)
- unite_metadata, [18](#)
- update_project_db, [18](#)