

# Package ‘chevreulPlot’

March 7, 2025

**Type** Package

**Title** Plots used in the chevreulPlot package

**Version** 0.99.34

**Description** Tools for plotting SingleCellExperiment objects in the chevreulPlot package. Includes functions for analysis and visualization of single-cell data. Supported by NIH grants R01CA137124 and R01EY026661 to David Cobrinik.

**License** MIT + file LICENSE

**URL** <https://github.com/whtns/chevreulPlot>,  
<https://whtns.github.io/chevreulPlot/>

**Date** 2024-11-17

**BugReports** <https://github.com/cobriniklab/chevreulPlot/issues>

**Depends** R (>= 4.5.0), SingleCellExperiment, chevreulProcess

**Imports** base, cluster, clustree, ComplexHeatmap (>= 2.5.4), circlize, dplyr, EnsDb.Hsapiens.v86, forcats, fs, ggplot2, grid, plotly, purrr, S4Vectors, scales, scater, scran, scuttle, stats, stringr, tibble, tidyr, utils, wigglyplotr (>= 1.13.1), tidyselect, patchwork

**Suggests** BiocStyle, knitr, RefManager, rmarkdown, testthat (>= 3.0.0)

**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/chevreulPlot>

**git\_branch** devel  
**git\_last\_commit** 5e39a4d  
**git\_last\_commit\_date** 2025-02-27  
**Repository** Bioconductor 3.21  
**Date/Publication** 2025-03-07  
**Author** Kevin Stachelek [aut, cre] (ORCID:  
<https://orcid.org/0000-0003-2085-695X>),  
 Bhavana Bhat [aut]  
**Maintainer** Kevin Stachelek <kevin.stachelek@gmail.com>

## Contents

|                                       |           |
|---------------------------------------|-----------|
| chevreulPlot-package . . . . .        | 2         |
| cc.genes.cyclone . . . . .            | 3         |
| enframe_markers . . . . .             | 4         |
| ensembl_version . . . . .             | 4         |
| grch38 . . . . .                      | 5         |
| grch38_tx2gene . . . . .              | 6         |
| human_to_mouse_homologs . . . . .     | 6         |
| make_complex_heatmap . . . . .        | 7         |
| plotly_settings . . . . .             | 8         |
| plot_all_transcripts . . . . .        | 9         |
| plot_colData_histogram . . . . .      | 9         |
| plot_colData_on_embedding . . . . .   | 10        |
| plot_feature_on_embedding . . . . .   | 11        |
| plot_gene_coverage_by_var . . . . .   | 12        |
| plot_marker_features . . . . .        | 13        |
| plot_transcript_composition . . . . . | 14        |
| plot_violin . . . . .                 | 15        |
| small_example_dataset . . . . .       | 16        |
| tiny_sce . . . . .                    | 16        |
| <b>Index</b>                          | <b>17</b> |

---

chevreulPlot-package    *chevreulPlot: Plots used in the chevreulPlot package*

---

## Description

Tools for plotting SingleCellExperiment objects in the chevreulPlot package. Includes functions for analysis and visualization of single-cell 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/chevreulPlot>
- <https://whtns.github.io/chevreulPlot/>
- Report bugs at <https://github.com/cobriniklab/chevreulPlot/issues>

---

cc.genes.cyclone

*Cyclone cell cycle pairs by symbol*

---

### 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

---

|                 |                                |
|-----------------|--------------------------------|
| enframe_markers | <i>Enframe Cluster Markers</i> |
|-----------------|--------------------------------|

---

**Description**

Enframe Cluster Markers

**Usage**

```
enframe_markers(tbl)
```

**Arguments**

tbl                    a tibble of marker genes

**Value**

a pivoted tibble of marker genes

---

|                 |                                       |
|-----------------|---------------------------------------|
| 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
```

---

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](http://ensembl.org/homo_sapiens)

## Examples

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

---

`grch38_tx2gene`*Human transcripts to genes*

---

**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](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

---

make\_complex\_heatmap *Plot Annotated Complexheatmap from SingleCellExperiment object*

---

**Description**

Plot Annotated Complexheatmap from SingleCellExperiment object

**Usage**

```
make_complex_heatmap(
  object,
  features = NULL,
  group.by = "ident",
  cells = NULL,
  assayName = "logcounts",
  experiment = NULL,
  group.bar.height = 0.01,
  column_split = NULL,
  col_arrangement = "ward.D2",
  mm_col_dend = 30,
  ...
)
```

**Arguments**

|                  |   |
|------------------|---|
| object           | A SingleCellExperiment object   |
| features         | Vector of features to plot. Features can come   |
| group.by         | Name of one or more metadata columns to annotate columns by (for example, orig.ident) |
| cells            | Cells to retain   |
| assayName        | "counts" for raw data "scale.data" for log-normalized data                            |
| experiment       | experiment to display   |
| group.bar.height | height for group bars   |
| column_split     | whether to split columns by metadata value  |

|                 |   |
|-----------------|---|
| col_arrangement | how to arrange columns whether with a dendrogram (Ward.D2, average, etc.) or exclusively by metadata category |
| mm_col_dend     | height of column dendrogram   |
| ...             | additional arguments passed to Heatmap  |

**Value**

a complexheatmap

**Examples**

```
data("tiny_sce")
make_complex_heatmap(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 altered settings



---

plot\_all\_transcripts *Plot All Transcripts*

---

**Description**

plot expression all transcripts for an input gene superimposed on embedding

**Usage**

```
plot_all_transcripts(  
  object,  
  features,  
  embedding = "UMAP",  
  from_gene = TRUE,  
  ...  
)
```

**Arguments**

|           |  |
|-----------|--|
| object    | A object   |
| features  | gene or vector of transcripts                            |
| embedding | umap   |
| from_gene | whether to look up transcripts for an input gene         |
| ...       | additional arguments passed to plot_feature_on_embedding |

**Value**

a list of embedding plots colored by a feature of interest

**Examples**

```
data("tiny_sce")  
plot_all_transcripts(tiny_sce, "NRL", from_gene = TRUE)
```

---

plot\_colData\_histogram

*Plot Read Count*

---

**Description**

Draw a box plot for read count data of a metadata variable

**Usage**

```
plot_colData_histogram(
  object,
  group_by = NULL,
  fill_by = NULL,
  yscale = "linear",
  return_plotly = FALSE
)
```

**Arguments**

|               |  |
|---------------|--|
| object        | A object   |
| group_by      | Metadata variable to plot. Default set to "nCount_RNA" |
| fill_by       | Variable to color bins by. Default set to "batch"      |
| yscale        | Scale of y axis. Default set to "linear"               |
| return_plotly | whether to return an interactive plotly plot           |

**Value**

a histogram of read counts

**Examples**

```
data(small_example_dataset)
small_example_dataset <- sce_calcn(small_example_dataset)
# static plot
plot_colData_histogram((small_example_dataset), return_plotly = FALSE)
```

---

plot\_colData\_on\_embedding

*Plot Metadata Variables*

---

**Description**

Plots static or interactive plot where each point represents a cell metadata variable whose position on the map depends on cell embeddings determined by the reduction technique used

**Usage**

```
plot_colData_on_embedding(
  object,
  group = "batch",
  embedding = "UMAP",
  dims = c(1, 2),
  highlight = NULL,
  return_plotly = FALSE,
  ...
)
```

**Arguments**

|               |  |
|---------------|--|
| object        | A SingleCellExperiment object                                  |
| group         | Name of one or more metadata columns to group (color) cells by |
| embedding     | The dimensional reduction technique to be used                 |
| dims          | Dimensions to plot, must be a two-length numeric vector        |
| highlight     | A list of vectors of cells to highlight                        |
| return_plotly | Convert plot to interactive web-based graph                    |
| ...           | extra parameters passed to ggplot                              |

**Value**

a ggplot

**Examples**

```
data(small_example_dataset)

# static mode
plot_colData_on_embedding(small_example_dataset, "Mutation_Status", return_plotly = FALSE)
```

---

```
plot_feature_on_embedding
      Plot Feature
```

---

**Description**

Plots gene or transcript expression overlaid on a given embedding.

**Usage**

```
plot_feature_on_embedding(
  object,
  embedding = c("UMAP", "PCA", "TSNE"),
  features,
  dims = c(1, 2),
  return_plotly = FALSE,
  ...
)
```

**Arguments**

|               |   |
|---------------|---|
| object        | A SingleCellExperiment object                           |
| embedding     | Dimensional reduction technique to be used              |
| features      | Feature to plot   |
| dims          | Dimensions to plot, must be a two-length numeric vector |
| return_plotly | return plotly object                                    |
| ...           | additional arguments passed to plotReduceDim            |

**Value**

an embedding colored by a feature of interest

**Examples**

```
data(small_example_dataset)
plot_feature_on_embedding(small_example_dataset, embedding = "UMAP",
  features = "Gene_0001")
```

---

```
plot_gene_coverage_by_var
```

*Plot BigWig Coverage for Genes of Interest by a Given Variable*

---

**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 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

---

plot\_marker\_features *Plot Cluster Marker Genes*

---

**Description**

Plot a dot plot of n marker features grouped by cell metadata available methods are wilcoxon rank-sum test

**Usage**

```
plot_marker_features(
  object,
  group_by = "batch",
  num_markers = 5,
  selected_values = NULL,
  return_plotly = FALSE,
  marker_method = "wilcox",
  experiment = "gene",
  hide_technical = NULL,
  unique_markers = FALSE,
  p_val_cutoff = 1,
  ...
)
```

**Arguments**

|                 |   |
|-----------------|---|
| object          | a object  |
| group_by        | the metadata variable from which to pick clusters   |
| num_markers     | default is 5  |
| selected_values | selected values to display                          |
| return_plotly   | whether to return an interactive plotly plot        |
| marker_method   | "wilcox"  |
| experiment      | experiment to plot default gene                     |
| hide_technical  | whether to exclude mitochondrial or ribosomal genes |
| unique_markers  | whether to plot only unique marker genes for group  |
| p_val_cutoff    | cutoff for p value display                          |
| ...             | extra parameters passed to ggplot2                  |

**Value**

a ggplot with marker genes from group\_by

**Examples**

```
data(small_example_dataset)
plot_marker_features(small_example_dataset, group_by = "gene_snn_res.1")
```

---

plot\_transcript\_composition  
*Plot Transcript Composition*

---

**Description**

plot the proportion of reads of a given gene map to each transcript

**Usage**

```
plot_transcript_composition(  
  object,  
  gene_symbol,  
  group.by = "batch",  
  standardize = FALSE,  
  drop_zero = FALSE  
)
```

**Arguments**

|             |   |
|-------------|---|
| object      | A object  |
| gene_symbol | Gene symbol of gene of interest   |
| group.by    | Name of one or more metadata columns to annotate columns by (for example, orig.ident) |
| standardize | whether to standardize values   |
| drop_zero   | Drop zero values  |

**Value**

a stacked barplot of transcript counts

**Examples**

```
data(tiny_sce)
plot_transcript_composition(tiny_sce, "NRL")
```

---

|             |                         |
|-------------|-------------------------|
| plot_violin | <i>Plot Violin plot</i> |
|-------------|-------------------------|

---

**Description**

Plots a Violin plot of a single data (gene expression, metrics, etc.) grouped by a metadata variable

**Usage**

```
plot_violin(
  object,
  group_by = "batch",
  plot_vals = NULL,
  features = "NRL",
  experiment = "gene",
  ...
)
```

**Arguments**

|            |  |
|------------|--|
| object     | A SingleCellExperiment object                            |
| group_by   | Variable to group (color) cells by                       |
| plot_vals  | plot values  |
| features   | Features to plot   |
| experiment | Name of experiment to use, defaults to active experiment |
| ...        | extra parameters passed to ggplot2                       |

**Value**

a violin plot

**Examples**

```
data("tiny_sce")
plot_violin(tiny_sce, "Prep.Method", features = "NRL")
```

---

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`

---

tiny\_sce *Tiny example SingleCellExperiment*

---

**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()`



# Index

## \* datasets

- cc.genes.cyclone, 3
- ensembl\_version, 4
- grch38, 5
- grch38\_tx2gene, 6
- human\_to\_mouse\_homologs, 6
- small\_example\_dataset, 16
- tiny\_sce, 16

## \* internal

- chevreulPlot-package, 2

- cc.genes.cyclone, 3
- chevreulPlot (chevreulPlot-package), 2
- chevreulPlot-package, 2

- enframe\_markers, 4
- ensembl\_version, 4

- grch38, 5
- grch38\_tx2gene, 6

- human\_to\_mouse\_homologs, 6

- make\_complex\_heatmap, 7

- plot\_all\_transcripts, 9
- plot\_colData\_histogram, 9
- plot\_colData\_on\_embedding, 10
- plot\_feature\_on\_embedding, 11
- plot\_gene\_coverage\_by\_var, 12
- plot\_marker\_features, 13
- plot\_transcript\_composition, 14
- plot\_violin, 15
- plotly\_settings, 8

- small\_example\_dataset, 16

- tiny\_sce, 16