

# Package ‘SingleCellComplexHeatMap’

May 7, 2026

**Type** Package

**Title** Complex Heatmaps for Single Cell Expression Data with Dual Information Display

**Version** 0.1.2

**Maintainer** XueCheng Fang <xuecheng328@gmail.com>

**Description** Creates complex heatmaps for single cell RNA-seq data that simultaneously display gene expression levels (as color intensity) and expression percentages (as circle sizes). Supports gene grouping, cell type annotations, and time point comparisons. Built on top of 'ComplexHeatmap' and integrates with 'Seurat' objects. For more details see Gu (2022) <[doi:10.1002/imt.2.43](https://doi.org/10.1002/imt.2.43)> and Hao (2024) <[doi:10.1038/s41587-023-01767-y](https://doi.org/10.1038/s41587-023-01767-y)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Depends** R (>= 4.0.0)

**Imports** ComplexHeatmap (>= 2.10.0), Seurat (>= 4.0.0), dplyr (>= 1.0.0), tidyr (>= 1.0.0), RColorBrewer, circlize (>= 0.4.0), grid, grDevices, stats, magrittr

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, viridis, devtools, BiocManager, ggsci, SeuratObject

**URL** <https://github.com/FanXuRong/SingleCellComplexHeatMap>

**BugReports** <https://github.com/FanXuRong/SingleCellComplexHeatMap/issues>

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** XueCheng Fang [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-06-28 17:40:02 UTC

## Contents

create_cell_annotations . . . . .	2
create_gene_annotations . . . . .	4
create_single_cell_complex_heatmap . . . . .	5
prepare_expression_matrices . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

create\_cell\_annotations  
*Create Cell Type and Time Point Annotations for Heatmap Columns*

---

### Description

Parses column names to extract time points and cell types, creates annotations and reorders matrices.

### Usage

```
create_cell_annotations(
  exp_mat,
  percent_mat,
  split_pattern = "_",
  time_position = 1,
  celltype_start = 2,
  time_points_order = NULL,
  cell_types_order = NULL,
  time_color_palette = "Accent",
  celltype_color_palette = "Dark2",
  show_time_annotation = TRUE,
  show_celltype_annotation = TRUE,
  time_point_title = "Time Point",
  cell_type_title = "Cell Type"
)
```

### Arguments

exp_mat	Expression matrix with samples as columns
percent_mat	Percentage matrix with samples as columns
split_pattern	Character string used to split column names (default: "_")
time_position	Integer indicating position of time point in split names (default: 1)
celltype_start	Integer indicating starting position of cell type in split names (default: 2)
time_points_order	Character vector specifying order of time points (default: NULL for automatic)
cell_types_order	Character vector specifying order of cell types (default: NULL for automatic)

time\_color\_palette  
Character string specifying palette name OR character vector of colors for time points (default: "Accent")

celltype\_color\_palette  
Character string specifying palette name OR character vector of colors for cell types (default: "Dark2")

show\_time\_annotation  
Logical indicating whether to show time point annotation (default: TRUE)

show\_celltype\_annotation  
Logical indicating whether to show cell type annotation (default: TRUE)

time\_point\_title  
Character string for time point annotation title (default: "Time Point")

cell\_type\_title  
Character string for cell type annotation title (default: "Cell Type")

**Value**

A list containing `exp_mat_ordered` (reordered expression matrix), `percent_mat_ordered` (reordered percentage matrix), `col_annotation` (ComplexHeatmap column annotation object), `col_split_factor` (factor for column splitting based on time points), and `annotation_df` (data frame with column annotations).

**See Also**

[create\\_single\\_cell\\_complex\\_heatmap](#), [prepare\\_expression\\_matrices](#)

**Examples**

```
# Load a small example Seurat object
data("pbmc_small", package = "SeuratObject")
pbmc_small$timepoint <- sample(c("0h", "6h"), ncol(pbmc_small), replace = TRUE)
pbmc_small$timepoint_celltype <- paste(pbmc_small$timepoint, pbmc_small$RNA_snn_res.0.8, sep = "_")
features <- c("CD3D", "CD79A", "MS4A1")

# Prepare expression matrices first
matrices <- prepare_expression_matrices(pbmc_small, features, group_by = "timepoint_celltype")

# Create cell annotations with custom ordering
col_annotations <- create_cell_annotations(
  exp_mat = matrices$exp_mat,
  percent_mat = matrices$percent_mat,
  split_pattern = "_",
  time_points_order = c("0h", "6h"),
  cell_types_order = levels(pbmc_small$RNA_snn_res.0.8)
)

# Access results
ordered_exp_mat <- col_annotations$exp_mat_ordered
```

---

`create_gene_annotations`*Create Gene Group Annotations for Heatmap Rows*

---

## Description

Creates gene grouping annotations and reorders expression matrices based on gene classifications.

## Usage

```
create_gene_annotations(  
  exp_mat,  
  percent_mat,  
  gene_classification,  
  color_palette = "Set1",  
  sort_within_groups = TRUE,  
  annotation_title = "Gene Group"  
)
```

## Arguments

<code>exp_mat</code>	Expression matrix with genes as rows
<code>percent_mat</code>	Percentage matrix with genes as rows
<code>gene_classification</code>	Named list where names are group labels and values are character vectors of gene names
<code>color_palette</code>	Character string specifying palette name OR character vector of colors (default: "Set1")
<code>sort_within_groups</code>	Logical indicating whether to sort genes within each group (default: TRUE)
<code>annotation_title</code>	Character string for annotation title (default: "Gene Group")

## Value

A list containing `exp_mat_ordered` (reordered expression matrix), `percent_mat_ordered` (reordered percentage matrix), `row_annotation` (ComplexHeatmap row annotation object), `row_split_factor` (factor for row splitting), and `annotation_df` (data frame with gene annotations).

## See Also

[create\\_single\\_cell\\_complex\\_heatmap](#), [prepare\\_expression\\_matrices](#)

## Examples

```
# Load a small example Seurat object
data("pbmc_small", package = "SeuratObject")
features <- c("CD3D", "CD79A", "MS4A1", "GZMK", "CCL5")

# Prepare expression matrices first
matrices <- prepare_expression_matrices(pbmc_small, features, group_by = "RNA_snn_res.0.8")

# Define gene groups
gene_groups <- list(
  "T-cell Markers" = c("CD3D", "GZMK", "CCL5"),
  "B-cell Markers" = c("CD79A", "MS4A1")
)

# Create gene annotations
annotations <- create_gene_annotations(
  exp_mat = matrices$exp_mat,
  percent_mat = matrices$percent_mat,
  gene_classification = gene_groups,
  color_palette = "Set1"
)

# Access results
ordered_exp_mat <- annotations$exp_mat_ordered
```

---

```
create_single_cell_complex_heatmap
```

*Create Complex Heatmap for Single Cell Expression Data*

---

## Description

Creates a complex heatmap that displays both gene expression levels (as color intensity) and expression percentages (as circle sizes) for single cell RNA-seq data. This function provides extensive customization options while maintaining ease of use.

## Usage

```
create_single_cell_complex_heatmap(
  seurat_object,
  features,
  gene_classification = NULL,
  group_by = "seurat_clusters",
  idents = NULL,
  time_points_order = NULL,
  cell_types_order = NULL,
  color_range = c(-1, 0, 2),
  color_palette = NULL,
```

```
max_circle_size = 2,  
row_fontsize = 8,  
col_fontsize = 9,  
col_name_rotation = 90,  
row_title_fontsize = 10,  
col_title_fontsize = 10,  
show_heatmap_legend = TRUE,  
show_percentage_legend = TRUE,  
legend_side = "right",  
cell_border_color = "grey80",  
split_pattern = "_",  
gene_color_palette = "Set1",  
time_color_palette = "Accent",  
celltype_color_palette = "Dark2",  
show_gene_grouping = NULL,  
show_time_annotation = TRUE,  
show_celltype_annotation = TRUE,  
split_by = "time",  
merge_legends = TRUE,  
percentage_legend_title = "Expression %",  
percentage_legend_labels = c("0%", "25%", "50%", "75%", "100%"),  
percentage_breaks = NULL,  
return_data = FALSE,  
save_plot = NULL,  
plot_width = 10,  
plot_height = 8,  
plot_dpi = 300,  
assay = NULL,  
slot = "scale.data",  
cluster_cells = TRUE,  
cluster_features = TRUE,  
clustering_distance_rows = "euclidean",  
clustering_distance_cols = "euclidean",  
clustering_method_rows = "complete",  
clustering_method_cols = "complete",  
color_palette_main = c("blue", "white", "red"),  
annotation_colors = NULL,  
show_feature_names = TRUE,  
feature_names_gp = NULL,  
legend_title = "Expression",  
gene_group_title = "Gene Group",  
time_point_title = "Time Point",  
cell_type_title = "Cell Type",  
show_cell_borders = TRUE,  
show_column_annotation = TRUE,  
gene_name_mapping = NULL,  
...  
)
```

**Arguments**

seurat_object	A Seurat object containing single cell data
features	Character vector of gene names to plot
gene_classification	Named list where names are group labels and values are character vectors of gene names (default: NULL for no gene grouping)
group_by	Character string specifying the metadata column to group by (default: "seurat_clusters")
idents	Numeric or character vector specifying which cell groups to include (default: NULL for all)
time_points_order	Character vector specifying order of time points. Only affects display order, not data filtering (default: NULL for automatic)
cell_types_order	Character vector specifying order of cell types. Only affects display order, not data filtering (default: NULL for automatic)
color_range	Numeric vector specifying color mapping break points for expression values. Its length must match color_palette if color_palette is a vector. (default: c(-1, 0, 2))
color_palette	Character vector specifying colors for expression heatmap. Its length must match color_range. If NULL, a default palette (viridis or color_palette_main) is generated to match color_range length (default: NULL)
max_circle_size	Numeric specifying maximum circle radius in mm. This applies to the highest percentage value in percentage_breaks (default: 2)
row_fontsize	Numeric specifying row name font size (default: 8)
col_fontsize	Numeric specifying column name font size (default: 9)
col_name_rotation	Numeric specifying column name rotation angle (default: 90)
row_title_fontsize	Numeric specifying row title font size (default: 10)
col_title_fontsize	Numeric specifying column title font size (default: 10)
show_heatmap_legend	Logical indicating whether to show heatmap legend (default: TRUE)
show_percentage_legend	Logical indicating whether to show percentage legend (default: TRUE)
legend_side	Character string specifying legend position (default: "right")
cell_border_color	Character string specifying cell border color (default: "grey80")
split_pattern	Character string used to split column names for parsing (default: "_")
gene_color_palette	Character string specifying palette name OR character vector of colors for gene groups (default: "Set1")

`time_color_palette` Character string specifying palette name OR character vector of colors for time points (default: "Accent")

`celltype_color_palette` Character string specifying palette name OR character vector of colors for cell types (default: "Dark2")

`show_gene_grouping` Logical indicating whether to show gene grouping (default: TRUE if `gene_classification` provided)

`show_time_annotation` Logical indicating whether to show time point annotation (default: TRUE)

`show_celltype_annotation` Logical indicating whether to show cell type annotation (default: TRUE)

`split_by` Character string specifying how to split columns: "time", "celltype", or "none" (default: "time")

`merge_legends` Logical indicating whether to merge legends (default: TRUE)

`percentage_legend_title` Character string for percentage legend title (default: "Expression %")

`percentage_legend_labels` Character vector for percentage legend labels

`percentage_breaks` Numeric vector specifying actual percentage values corresponding to labels

`return_data` Logical; if TRUE, return underlying data instead of drawing only

`save_plot` File path to save the drawn heatmap (PNG)

`plot_width` Numeric; width in inches for saving

`plot_height` Numeric; height in inches for saving

`plot_dpi` Numeric; resolution (DPI) for saved plot

`assay` Seurat assay name to extract data from

`slot` Seurat slot name within assay (e.g., "scale.data", "data")

`cluster_cells` Logical; whether to cluster columns (cells)

`cluster_features` Logical; whether to cluster rows (features)

`clustering_distance_rows` Distance metric for row clustering

`clustering_distance_cols` Distance metric for column clustering

`clustering_method_rows` Clustering method for rows

`clustering_method_cols` Clustering method for columns

`color_palette_main` Fallback color palette when viridis unavailable

annotation\_colors      Named list of custom annotation colors  
 show\_feature\_names      Logical; whether to show feature (row) names  
 feature\_names\_gp      gpar object controlling feature name appearance  
 legend\_title      Character; title for main heatmap legend  
 gene\_group\_title      Character string for gene group annotation title (default: "Gene Group")  
 time\_point\_title      Character string for time point annotation title (default: "Time Point")  
 cell\_type\_title      Character string for cell type annotation title (default: "Cell Type")  
 show\_cell\_borders      Logical indicating whether to show cell border lines (default: TRUE)  
 show\_column\_annotation      Logical indicating whether to show column annotations (default: TRUE)  
 gene\_name\_mapping      Named character vector for mapping gene names, where names are original gene names and values are display names (default: NULL)  
 ...      Additional arguments passed to ComplexHeatmap::Heatmap()

**Value**

A ComplexHeatmap object. If return\_data is TRUE, returns a list containing the heatmap object and underlying data matrices.

---

```
prepare_expression_matrices
```

*Prepare Expression and Percentage Matrices from Seurat DotPlot*

---

**Description**

Extracts and reshapes expression data from a Seurat DotPlot object into matrices suitable for complex heatmap visualization.

**Usage**

```
prepare_expression_matrices(
  seurat_object,
  features,
  group_by = "seurat_clusters",
  idents = NULL,
  split_pattern = "_",
  time_position = 1,
  celltype_start = 2
)
```

**Arguments**

seurat_object	A Seurat object containing single cell data
features	Character vector of gene names to plot
group_by	Character string specifying the metadata column to group by (default: "seurat_clusters")
idents	Numeric or character vector specifying which cell groups to include (default: NULL for all)
split_pattern	Character string used to split column names for parsing (default: "_")
time_position	Integer indicating position of time point in split names (default: 1)
celltype_start	Integer indicating starting position of cell type in split names (default: 2)

**Value**

A list containing `exp_mat` (matrix of scaled expression values), `percent_mat` (matrix of expression percentages), and `dotplot_data` (original DotPlot data frame).

**See Also**

[create\\_single\\_cell\\_complex\\_heatmap](#)

**Examples**

```
# Load a small example Seurat object
data("pbmc_small", package = "SeuratObject")
features <- c("CD3D", "CD79A", "MS4A1")

# Basic usage
matrices <- prepare_expression_matrices(
  seurat_object = pbmc_small,
  features = features,
  group_by = "RNA_snn_res.0.8"
)

# Access the results
expression_matrix <- matrices$exp_mat
percentage_matrix <- matrices$percent_mat
```

# Index

`create_cell_annotations`, [2](#)  
`create_gene_annotations`, [4](#)  
`create_single_cell_complex_heatmap`, [3](#),  
[4](#), [5](#), [10](#)  
`prepare_expression_matrices`, [3](#), [4](#), [9](#)