

# Package ‘SimiCviz’

September 16, 2026

**Type** Package

**Title** Visualization Tools for Gene Regulatory Network Analysis

**Version** 0.99.2

**Description** Visualization and export utilities for SimiC and SimiCPipeline outputs. The package focuses on importing SimiC-style results (e.g., weights, AUC metrics) from pickle/CSV files, processing and generating publication-ready plots (networks, heatmaps, distributions) and tables saved into a reproducible, ordered directory hierarchy.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** false

**SystemRequirements** Python (>= 3.8)

**Depends** R (>= 4.6)

**Imports** methods, stats, BiocParallel, viridisLite, utils, Matrix, graphics, SummarizedExperiment, colorspace, gridExtra, ggplot2, dplyr, tibble, tidyr, reshape2, scales, reticulate (>= 1.45.0)

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

**VignetteBuilder** knitr

**URL** <https://github.com/ML4BM-Lab/SimiCviz>

**BugReports** <https://github.com/ML4BM-Lab/SimiCviz/issues>

**biocViews** Software, Visualization, GeneRegulation, NetworkInference, Network, SingleCell

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**git\_url** <https://git.bioconductor.org/packages/SimiCviz>

**git\_branch** devel

**git\_last\_commit** ba290d2

**git\_last\_commit\_date** 2026-07-04

**Repository** Bioconductor 3.24

**Date/Publication** 2026-09-15

**Author** Irene Marín-Goñi [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-5060-0712>>)

**Maintainer** Irene Marín-Goñi <imarin.4@alumni.unav.es>

## Contents

|                                           |    |
|-------------------------------------------|----|
| .compute_cell_auc . . . . .               | 3  |
| .ensure_genes_x_cells_format . . . . .    | 3  |
| .make_weight_barplot . . . . .            | 4  |
| .plot_weights_paginated . . . . .         | 4  |
| .subset_auc_for_label . . . . .           | 5  |
| .weights_to_long . . . . .                | 6  |
| AUCProcessor . . . . .                    | 6  |
| AUCProcessor-class . . . . .              | 7  |
| auc_list_to_df . . . . .                  | 8  |
| calculate_activity_scores . . . . .       | 9  |
| calculate_dissimilarity . . . . .         | 10 |
| calculate_ecdf_auc . . . . .              | 11 |
| compute_auc . . . . .                     | 12 |
| compute_auc,AUCProcessor-method . . . . . | 13 |
| export_SimiCviz_csv . . . . .             | 13 |
| get_auc . . . . .                         | 14 |
| get_auc,AUCProcessor-method . . . . .     | 15 |
| get_tf_network . . . . .                  | 15 |
| is.SimiCvizExperiment . . . . .           | 16 |
| load_cell_labels . . . . .                | 17 |
| load_collected_auc . . . . .              | 18 |
| load_expression_matrix . . . . .          | 18 |
| load_from_csv . . . . .                   | 19 |
| load_SimiCPipeline . . . . .              | 20 |
| plot_auc_cumulative . . . . .             | 21 |
| plot_auc_distributions . . . . .          | 22 |
| plot_auc_heatmap . . . . .                | 23 |
| plot_auc_summary_statistics . . . . .     | 24 |
| plot_dissimilarity_heatmap . . . . .      | 25 |
| plot_r2_distribution . . . . .            | 26 |
| plot_target_weights . . . . .             | 27 |
| plot_tf_network_heatmap . . . . .         | 28 |
| plot_tf_weights . . . . .                 | 30 |
| read_auc_csv . . . . .                    | 31 |
| read_auc_pickle . . . . .                 | 31 |
| read_pickle . . . . .                     | 32 |
| read_weights_csv . . . . .                | 32 |
| read_weights_pickle . . . . .             | 33 |
| setCellLabels . . . . .                   | 34 |
| setLabelNames . . . . .                   | 34 |
| SimiCviz . . . . .                        | 35 |
| SimiCvizExperiment . . . . .              | 36 |
| SimiCvizExperiment-class . . . . .        | 37 |

---

|                                |                                      |
|--------------------------------|--------------------------------------|
| <code>.compute_cell_auc</code> | <i>Compute AUC for a single cell</i> |
|--------------------------------|--------------------------------------|

---

**Description**

Compute AUC for a single cell

**Usage**

```
.compute_cell_auc(  
  expr_row,  
  weight_df,  
  target_norm,  
  sort_by,  
  select_top_k = NULL,  
  percent_of_target = 1  
)
```

**Arguments**

|                                |                                                                              |
|--------------------------------|------------------------------------------------------------------------------|
| <code>expr_row</code>          | expression values for target genes (vector)                                  |
| <code>weight_df</code>         | weight data.frame subset for this cell's label (columns: tf, target, weight) |
| <code>target_norm</code>       | precomputed Euclidean norm of target expression (vector)                     |
| <code>sort_by</code>           | sorting criterion ("expression" or "weight")                                 |
| <code>select_top_k</code>      | keep only top K targets per TF (NULL = all)                                  |
| <code>percent_of_target</code> | percentage of targets to use (0-1)                                           |

**Value**

named vector of AUC scores (one per TF)

---

|                                           |                                                              |
|-------------------------------------------|--------------------------------------------------------------|
| <code>.ensure_genes_x_cells_format</code> | <i>Validate expression matrix is in genes x cells format</i> |
|-------------------------------------------|--------------------------------------------------------------|

---

**Description**

Strictly enforces that expression matrix is in genes x cells format with column names matching cell identifiers in `cell_labels`.

**Usage**

```
.ensure_genes_x_cells_format(expr_mat, cell_labels)
```

**Arguments**

|                          |                                                   |
|--------------------------|---------------------------------------------------|
| <code>expr_mat</code>    | expression matrix                                 |
| <code>cell_labels</code> | data.frame with cell identifiers in 'cell' column |

**Value**

expression matrix (genes x cells) with validated dimensions

---

`.make_weight_barplot`     *Render a single weight barplot on a ggplot*

---

**Description**

Render a single weight barplot on a ggplot

**Usage**

```
.make_weight_barplot(
  df,
  gene_name,
  gene_type,
  col_map,
  top_n = NULL,
  allowed_genes = NULL
)
```

**Arguments**

|                            |                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <code>df</code>            | long-format data.frame (already subsetting) with columns: <code>gene</code> , <code>weight</code> , <code>label_name</code> |
| <code>gene_name</code>     | character                                                                                                                   |
| <code>gene_type</code>     | character; "tf" or "target"                                                                                                 |
| <code>col_map</code>       | named character vector of colours (keyed by <code>label_name</code> )                                                       |
| <code>top_n</code>         | integer or NULL; keep only top N partners by mean abs weight                                                                |
| <code>allowed_genes</code> | character vector or NULL; restrict partner genes to this set                                                                |

**Value**

ggplot object

---

`.plot_weights_paginated`  
*Shared paginated weight barplot engine*

---

**Description**

Shared paginated weight barplot engine

**Usage**

```
.plot_weights_paginated(
  x,
  gene_names = NULL,
  gene_type = c("tf", "target"),
  labels = NULL,
  top_n = NULL,
  allowed_targets = NULL,
  grid = c(4L, 1L),
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 16,
  height = NULL
)
```

**Arguments**

|                              |                                                                                                              |
|------------------------------|--------------------------------------------------------------------------------------------------------------|
| <code>x</code>               | SimiCvizExperiment                                                                                           |
| <code>gene_names</code>      | character vector or NULL (all genes of that type)                                                            |
| <code>gene_type</code>       | "tf" or "target"                                                                                             |
| <code>labels</code>          | integer vector of labels (default: all)                                                                      |
| <code>top_n</code>           | integer or NULL; max partners shown per gene (TF mode only)                                                  |
| <code>allowed_targets</code> | character vector or NULL; restrict targets to this set (TF mode only; use to pass pre-filtered target lists) |
| <code>grid</code>            | integer vector c(nrow, ncol) per page; NULL = single page                                                    |
| <code>save</code>            | logical                                                                                                      |
| <code>filename</code>        | PDF filename                                                                                                 |
| <code>out_dir</code>         | output directory                                                                                             |
| <code>width, height</code>   | page dimensions in inches                                                                                    |

**Value**

Invisibly, list of page grobs.

---

`.subset_auc_for_label` *Subset collected AUC for a specific label*

---

**Description**

Subset collected AUC for a specific label

**Usage**

```
.subset_auc_for_label(x, label)
```

---

|                               |                                                      |
|-------------------------------|------------------------------------------------------|
| <code>.weights_to_long</code> | <i>Convert weight list to long-format data frame</i> |
|-------------------------------|------------------------------------------------------|

---

### Description

Convert weight list to long-format data frame

### Usage

```
.weights_to_long(x)
```

### Arguments

|                |                                 |
|----------------|---------------------------------|
| <code>x</code> | <code>SimiCvizExperiment</code> |
|----------------|---------------------------------|

### Value

data.frame with columns: tf, target, weight, label, label\_name

---

|                           |                                |
|---------------------------|--------------------------------|
| <code>AUCProcessor</code> | <i>Initialize AUCProcessor</i> |
|---------------------------|--------------------------------|

---

### Description

Initialize AUCProcessor

### Usage

```
AUCProcessor(
  weights,
  expression,
  cell_labels,
  qc_type = NULL,
  qc_threshold = 0.7,
  adj_r2_list = NULL,
  n_cores = 1,
  backend = "sequential"
)
```

### Arguments

|                          |                                                                                                                                                                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>weights</code>     | list with weight matrices per label (if simic input) or data.frame with columns: tf, target, weight, label, tf (if data.frame input). <code>adj_p_val</code> (optional, for p-value filtering <code>'qc_type' = "p_value"</code> ). |
| <code>expression</code>  | expression matrix (cells x cells) or path to file. <b>IMPORTANT:</b> Expression must be in genes x cells format (genes as rows, cells as columns), matching Seurat/SingleCellExperiment conventions.                                |
| <code>cell_labels</code> | data.frame with cell-to-label mapping                                                                                                                                                                                               |

|              |                                                                                                                                                                                                                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| qc_type      | character type of quality metric. Values: "adj_r2" or other column name (pvalue, adj_p_val, etc.) provided in weights in data.frame format. If "adj_r2", targets with R2 below threshold are filtered out. If other (i.e. "p_value"), targets with qc_metric above threshold are filtered out. |
| qc_threshold | threshold for filtering targets based on qc_metric (default: 0.7 for R2).                                                                                                                                                                                                                      |
| adj_r2_list  | (optional) A list of R2 values per label per Only used if weights are provided in list format qc_type is "adj_r2".                                                                                                                                                                             |
| n_cores      | number of cores for parallelization (default: 1)                                                                                                                                                                                                                                               |
| backend      | parallelization backend ("sequential", "multicore", "multisession")                                                                                                                                                                                                                            |

### Value

A [AUCProcessor](#) object with initialized slots.

### Examples

```
weight_path <- system.file("extdata", "example_weights.csv",
                           package = "SimiCviz")
weights_df <- read_weights_csv(weight_path)
expression_mat_path <- system.file("extdata",
                                   file.path("inputFiles", "example_expression.pickle"),
                                   package = "SimiCviz")
cell_labels_path <- system.file("extdata",
                                file.path("inputFiles", "disease_stage_annotation.csv"),
                                package = "SimiCviz")
cell_labels <- load_cell_labels(cell_labels_path, header = TRUE, sep = ",")

AS_processor <- AUCProcessor(weights = weights_df,
                             expression = expression_mat_path,
                             cell_labels = cell_labels,
                             qc_type = "adj_p_val",
                             qc_threshold = 0.05, # Filter targets above
                             n_cores = 2,
                             backend = "multisession")
```

---

|                    |                           |
|--------------------|---------------------------|
| AUCProcessor-class | <i>AUCProcessor class</i> |
|--------------------|---------------------------|

---

### Description

Efficient calculator for TF activity scores (AUC) based on GRN weights and cell-specific expression.

### Usage

```
## S4 method for signature 'AUCProcessor'
show(object)
```

### Arguments

object      A [AUCProcessor](#) object.

**Value**

An object of class AUCProcessor with slots for weights, expression, and computed AUC scores.

**Slots**

weights list with weight matrices per label (TFs x targets)  
 expression data.frame or matrix (genes x cells)  
 cell\_labels data.frame with columns 'cell' and 'label'  
 target\_ids character vector of target gene IDs  
 tf\_ids character vector of TF gene IDs  
 auc\_results matrix to store computed activity scores (cells x TFs)  
 parallel\_params list with parallelization settings

**Examples**

```
# Initialize an AUCProcessor object
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30)))),
                    "1" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30)))))
cell_labels <- data.frame(cell= c("Cell1", "Cell2"), label = c("0", "1"))
expr <- matrix(rnorm(60), nrow=2, byrow = FALSE,
              dimnames = list(c("Cell1", "Cell2"),paste0("Gene", 1:30)))

processor <- AUCProcessor(weights = weights_list,
                        expression = expr,
                        cell_labels = cell_labels)
```

---

auc\_list\_to\_df

*Collect per-label AUC matrices using cell labels*

---

**Description**

Given a list of AUC matrices (one per label) and a cell-labels data.frame, subsets each matrix to include only the cells belonging to that label and collects them into a single data.frame. This is a helper function to transform raw per-label AUC outputs into the "collected" format used by SimiCvizExperiment@auc.

**Usage**

```
auc_list_to_df(auc_list, cell_labels_df)
```

**Arguments**

auc\_list            Named list of AUC matrices (cells × TFs). Names should correspond to label identifiers (e.g. "0", "1").

cell\_labels\_df    A data.frame with columns cell and label, as returned by [load\\_cell\\_labels](#).

**Value**

A data.frame with cells in rows and TFs in columns, containing the activity scores for the specific labels.

**Examples**

```
# Example usage with per-label AUC matrices
auc_list <- list("0" = matrix(rnorm(100), nrow = 10,
                             dimnames = list(paste0("cell_", 1:10),
                                             paste0("TF", 1:10))),
               "1" = matrix(rnorm(100), nrow = 10,
                             dimnames = list(paste0("cell_", 1:10),
                                             paste0("TF", 1:10))))
cell_labels_df <- data.frame(cell = paste0("cell_", 1:10),
                             label = c(rep(0, 5), rep(1, 5)))
collected_auc <- auc_list_to_df(auc_list, cell_labels_df)
```

---

calculate\_activity\_scores

*Calculate activity scores from SimiCvizExperiment weights*

---

**Description**

Efficiently computes TF activity scores for all cells using label-specific GRNs. Avoids redundant computation by calculating only for each cell's corresponding label.

**Usage**

```
calculate_activity_scores(
  simic,
  expression = NULL,
  adj_r2_threshold = NULL,
  sort_by = "expression",
  select_top_k = NULL,
  percent_of_target = 1,
  n_cores = 1,
  backend = "sequential",
  verbose = TRUE
)
```

**Arguments**

|                  |                                                                                                |
|------------------|------------------------------------------------------------------------------------------------|
| simic            | SimiCvizExperiment object with weights and expression data                                     |
| expression       | expression matrix (cells x genes). If NULL, uses data from metadata or file path if available. |
| adj_r2_threshold | minimum adjusted R2 for target filtering (default: 0.7)                                        |
| sort_by          | sorting criterion ("expression", "weight", or "adj_r2")                                        |
| select_top_k     | keep only top K targets per TF (NULL = all)                                                    |

|                   |                                                                     |
|-------------------|---------------------------------------------------------------------|
| percent_of_target | percentage of targets to use (0-1)                                  |
| n_cores           | number of cores for parallelization (1 = sequential)                |
| backend           | parallelization backend ("sequential", "multicore", "multisession") |
| verbose           | print progress messages                                             |

**Value**

SimiCvizExperiment with computed AUC scores added to @auc slot

**Examples**

```
# Generate minimal example data
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30)))),
                    "1" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30))))))
cell_labels <- data.frame(cell= c("Cell1", "Cell2"), label = c("0", "1"))
simic <- SimiCvizExperiment(weights = weights_list, cell_labels = cell_labels)
expr <- matrix(rnorm(60), nrow=2, byrow = FALSE,
              dimnames = list(c("Cell1", "Cell2"),paste0("Gene", 1:30)))
simic <- calculate_activity_scores(simic,
                                expression = expr,
                                sort_by = "expression",
                                n_cores = 2,
                                backend ="multicore")
```

---

calculate\_dissimilarity

*Calculate dissimilarity scores across phenotype labels*

---

**Description**

Compares per-cell TF activity-score distributions between phenotype labels using a histogram-based MinMax divergence.

**Usage**

```
calculate_dissimilarity(
  x,
  labels = NULL,
  cell_groups = NULL,
  n_breaks = 100L,
  verbose = TRUE
)
```

**Arguments**

|             |                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------|
| x           | A SimiCvizExperiment object.                                                                                                       |
| labels      | Integer vector of label keys or character vector of label display names to compare (default: all labels). Mixing types is allowed. |
| cell_groups | Optional named list mapping group names to character vectors of cell identifiers.                                                  |
| n_breaks    | Number of histogram bins (default 100).                                                                                            |
| verbose     | Logical; print progress messages (default TRUE).                                                                                   |

**Value**

A data.frame with TFs as rows, sorted by score (descending).

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
dis_score <- calculate_dissimilarity(simic)
```

---

|                    |                           |
|--------------------|---------------------------|
| calculate_ecdf_auc | <i>Calculate ECDF AUC</i> |
|--------------------|---------------------------|

---

**Description**

Calculates the area under the empirical cumulative distribution function.

**Usage**

```
calculate_ecdf_auc(
  x,
  tf_names = NULL,
  labels = NULL,
  integration_range = c(0, 1),
  percentile = 0.5
)
```

**Arguments**

|                   |                                                                                 |
|-------------------|---------------------------------------------------------------------------------|
| x                 | A SimiCvizExperiment object.                                                    |
| tf_names          | Optional TF identifiers to evaluate. Defaults to all TFs in collected AUC data. |
| labels            | Optional labels to evaluate. Accepts label ids or label names.                  |
| integration_range | Numeric vector of length 2 defining lower and upper bounds for integration.     |
| percentile        | Percentile used to compute the auxiliary AUC50/x_at_p50 metrics.                |

**Value**

A data.frame with one row per TF and per-label ECDF-derived metrics.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds", package = "SimiCviz"))
ecdf_metrics <- calculate_ecdf_auc(simic, tf_names = c("MEF2D", "SATB1"))
head(ecdf_metrics)
```

---

compute\_auc

*Compute activity scores for an object*


---

**Description**

S4 generic for computing TF activity scores (AUC-like metrics).

**Usage**

```
compute_auc(object, ...)
```

**Arguments**

**object**            An object that implements a compute\_auc method.

**...**            Additional method-specific arguments.

**Value**

Object with computed activity scores.

**Examples**

```
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
  dimnames = list(paste0("TF", 1:3),
    paste0("Gene", 1:30)))),
  "1" = as.data.frame(matrix(rnorm(90), nrow = 3,
  dimnames = list(paste0("TF", 1:3),
    paste0("Gene", 1:30))))))
cell_labels <- data.frame(cell= c("Cell1", "Cell2"), label = c("0", "1"))
expr <- matrix(rnorm(60), nrow=2, byrow = FALSE,
  dimnames = list(c("Cell1", "Cell2"), paste0("Gene", 1:30)))

processor <- AUCProcessor(weights = weights_list,
  expression = expr,
  cell_labels = cell_labels)
processor <- compute_auc(processor, sort_by = "expression")
```

---

compute\_auc, AUCProcessor-method

*Compute activity scores for all cells*


---

### Description

Efficiently calculates TF activity scores for each cell using its corresponding label-specific GRN. Uses BiocParallel for parallelization.

### Usage

```
## S4 method for signature 'AUCProcessor'
compute_auc(
  object,
  sort_by = "expression",
  select_top_k = NULL,
  percent_of_target = 1,
  verbose = TRUE
)
```

### Arguments

|                   |                                                 |
|-------------------|-------------------------------------------------|
| object            | A <a href="#">AUCProcessor</a> object.          |
| sort_by           | sorting criterion ("expression" or "weight")    |
| select_top_k      | keep only top K targets per TF (NULL = use all) |
| percent_of_target | percentage of targets to use (0-1)              |
| verbose           | print progress messages                         |

### Value

A [AUCProcessor](#) object with computed scores in the auc\_results slot.

---

export\_SimiCviz\_csv    *Export SimiCvizExperiment tables to CSV*


---

### Description

Exports weights and AUC tables into an organized directory structure:

**data/weights** weights tables

**data/auc** AUC tables (collected cells × TF format)

### Usage

```
export_SimiCviz_csv(x, out_dir, prefix = "SimiCviz", overwrite = FALSE)
```

**Arguments**

x [SimiCvizExperiment](#) object.  
 out\_dir root output directory.  
 prefix filename prefix (default: "SimiCviz").  
 overwrite logical; overwrite existing files.

**Value**

Invisibly, a list of file paths.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
export_SimiCviz_csv(simic, out_dir = tempdir(), prefix = "simic_export")
```

---

|         |                                               |
|---------|-----------------------------------------------|
| get_auc | <i>Extract activity scores from an object</i> |
|---------|-----------------------------------------------|

---

**Description**

S4 generic for retrieving computed TF activity scores.

**Usage**

```
get_auc(object, ...)
```

**Arguments**

object An object that implements a get\_auc method.  
 ... Additional method-specific arguments.

**Value**

Extracted activity scores.

**Examples**

```
#' # Generate minimal example data
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30)))),
                    "1" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30))))))
cell_labels <- data.frame(cell= c("Cell1", "Cell2"), label = c("0", "1"))
set.seed(123)
expr <- matrix(rnorm(60), nrow=2, byrow = FALSE,
               dimnames = list(c("Cell1", "Cell2"), paste0("Gene", 1:30)))

processor <- AUCProcessor(weights = weights_list,
                          expression = expr,
```

```

                                cell_labels = cell_labels)
processor <- compute_auc(processor, sort_by = "expression")
auc_scores <- get_auc(processor)
head(auc_scores)
#           TF1           TF2           TF3
# Cell11 0.5308869 0.5307807 0.5618426
# Cell12 0.5752393 0.5399460 0.5304443

```

---

get\_auc,AUCProcessor-method

*Get computed activity scores*


---

## Description

Get computed activity scores

## Usage

```

## S4 method for signature 'AUCProcessor'
get_auc(object, format = "wide")

```

## Arguments

|        |                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------|
| object | A <a href="#">AUCProcessor</a> object.                                                                         |
| format | Output format: "wide" for cells x TF table, or "long" for long format with columns cell, tf, score, and label. |

## Value

A data.frame with activity scores in the requested format.

---

get\_tf\_network

*Extract stacked TF-target weight matrix across labels*


---

## Description

For a given TF, extracts the regulatory weight for every target gene in each label. Targets whose adjusted  $R^2$  falls below `r2_threshold` in a given label receive NA (analogous to Python's `get_TF_network(..., stacked=TRUE)`).

## Usage

```
get_tf_network(x, tf_name, labels = NULL, r2_threshold = NULL)
```

## Arguments

|              |                                                                                                       |
|--------------|-------------------------------------------------------------------------------------------------------|
| x            | A <code>SimiCvizExperiment</code> object.                                                             |
| tf_name      | Character; name of the transcription factor.                                                          |
| labels       | Integer vector of labels to include (default: all).                                                   |
| r2_threshold | Numeric or NULL. If non-NULL, weights for targets with adjusted $R^2$ below this value are set to NA. |

**Value**

A data.frame with target genes as rows and one column per label (named by label\_names).

**Examples**

```
# Example usage
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
network <- get_tf_network(simic, c("MEF2D"),
                          labels = c(0, 1),
                          r2_threshold = 0.7)

print(network)
```

---

is.SimiCvizExperiment *Check if object is a SimiCvizExperiment*

---

**Description**

Tests whether an object is of class SimiCvizExperiment.

**Usage**

```
is.SimiCvizExperiment(x)
```

**Arguments**

x                      An object to test.

**Value**

Logical, TRUE if x is a SimiCvizExperiment.

**Examples**

```
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30)))),
                    "1" = as.data.frame(matrix(rnorm(90), nrow = 3),
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30))))

simic <- SimiCvizExperiment(weights = weights_list)
is.SimiCvizExperiment(simic) # should return TRUE
```

---

|                  |                                   |
|------------------|-----------------------------------|
| load_cell_labels | <i>Load cell-to-label mapping</i> |
|------------------|-----------------------------------|

---

## Description

Reads or constructs a two-column `data.frame` (`cell`, `label`) from various input formats.

## Usage

```
load_cell_labels(x, ...)
```

## Arguments

`x` One of:

- character (length 1)** Path to a CSV / TSV file. If the file has columns named `cell` and `label` they are used directly. Otherwise the first column is treated as `cell` and the second as `label`. A single-column file (or headerless file with one field) is treated as a label-only vector — see next bullet.
- character / integer / numeric vector** A vector of labels, one per cell, in the **same order** as rows of the AUC matrix. Cell IDs are generated as `cell_1`, `cell_2`, ....
- named vector** Names are used as cell IDs and values as labels.
- data.frame** Must contain columns `cell` and `label`, or exactly two columns (first = `cell`, second = `label`).

`...` Additional arguments passed to file readers when `x` is a path.

## Value

A `data.frame` with columns `cell` (character) and `label` (integer), sorted by `cell`.

## Examples

```
cell_labels_path <- system.file("extdata",
                                file.path("inputFiles", "disease_stage_annotation.csv"),
                                package = "SimiCviz")
cell_labels <- load_cell_labels(cell_labels_path, header = TRUE, sep = ",")
head(cell_labels)
#           cell category label
# 1 AGGGTGATCTGAGGGA-1-NBM-10.138P      NBM      0
# 2 CCTAGCTTCTCCAACC-1-NBM-1.138P      NBM      0
# 3 ATTACTCTCGTGGTCG-1-NBM-10.138P      NBM      0
# 4 TAAGAGAAGCCGCCTA-1-NBM-8.138P      NBM      0
# 5 GATCGATTGAGAGGTG-1-NBM-8.138P      NBM      0
# 6 ACATCAGGTCGCGGTT-1-NBM-10.138P      NBM      0
```

---

|                    |                                                   |
|--------------------|---------------------------------------------------|
| load_collected_auc | <i>Load a collected AUC CSV into a data.frame</i> |
|--------------------|---------------------------------------------------|

---

### Description

Reads a cells  $\times$  TF CSV file (as written by internal function `.save_collected_auc` or `SimiCPipeline`). This is the canonical import for any GRN method that produces per-cell TF activity scores.

### Usage

```
load_collected_auc(file, ...)
```

### Arguments

|      |                                                           |
|------|-----------------------------------------------------------|
| file | Path to a CSV file with cells in rows and TFs in columns. |
| ...  | Additional arguments passed to <a href="#">read.csv</a> . |

### Value

A data.frame with cell IDs as row names.

### Examples

```
auc_file <- system.file("extdata",
  file.path("outputSimic/example_simic_auc_collected.csv"),
  package = "SimiCviz")
auc_df <- load_collected_auc(auc_file)
```

---

load\_expression\_matrix

*Load expression matrix from various sources /file formats Allowed formats:*

- *CSV file*
  - *h5ad file*
  - *rds file (Seurat or SingleCellExperiment object)*
  - *matrix or data.frame in R*
- 

### Description

Load expression matrix from various sources /file formats Allowed formats:

- CSV file
- h5ad file
- rds file (Seurat or SingleCellExperiment object)
- matrix or data.frame in R

**Usage**

```
load_expression_matrix(expression)
```

**Arguments**

expression      file path, matrix, or data.frame

**Value**

expression matrix

**Examples**

```
expression_mat_path <- system.file("extdata",
                                   file.path("inputFiles", "example_expression.pickle"),
                                   package = "SimiCviz")
expression_mat <- load_expression_matrix(expression_mat_path)
```

---

|               |                                                   |
|---------------|---------------------------------------------------|
| load_from_csv | <i>Create a SimiCvizExperiment from CSV files</i> |
|---------------|---------------------------------------------------|

---

**Description**

Create a SimiCvizExperiment from CSV files

**Usage**

```
load_from_csv(
  weights_file,
  auc_file = NULL,
  cell_labels_file = NULL,
  meta = list()
)
```

**Arguments**

weights\_file      CSV with weights (long format: tf, target, weight, and optionally label).  
 auc\_file          optional CSV with AUC metrics (collected cells × TF format).  
 cell\_labels\_file      optional CSV with cell labels (required columns: cell, label).  
 meta              optional named list with metadata.

**Value**

[SimiCvizExperiment](#) object.

## Examples

```
# Example usage with CSV files
weight_path <- system.file("extdata", "example_weights.csv",
                           package = "SimiCviz")
auc_path <- system.file("extdata", "example_auc.csv",
                        package = "SimiCviz")
cell_labels_path <- system.file("extdata",
                                file.path("inputFiles", "disease_stage_annotation.csv"),
                                package = "SimiCviz")
simic <- load_from_csv(weights_file = weight_path,
                      auc_file = auc_path,
                      cell_labels_file = cell_labels_path,
                      meta = list(run_name = "example_run"))
is.SimiCvizExperiment(simic) # TRUE
```

---

|                    |                                                            |
|--------------------|------------------------------------------------------------|
| load_SimiCPipeline | Create a SimiCvizExperiment from SimiCPipeline output path |
|--------------------|------------------------------------------------------------|

---

## Description

Loads weights and AUC data following this priority chain for AUC:

1. Collected CSV (cells  $\times$  TF, all labels merged) load\_collected\_auc
2. Per-label pickle  $\rightarrow$  collect  $\rightarrow$  save CSV read\_auc\_pickle  $\rightarrow$  auc\_list\_to\_df  $\rightarrow$  .save\_collected\_auc (internal)

## Usage

```
load_SimiCPipeline(
  project_dir,
  run_name = NULL,
  lambda1 = NULL,
  lambda2 = NULL,
  meta = list()
)
```

## Arguments

|             |                                            |
|-------------|--------------------------------------------|
| project_dir | root directory of a SimiCPipeline project. |
| run_name    | name of the run experiment.                |
| lambda1     | L1 regularization parameter.               |
| lambda2     | L2 regularization parameter.               |
| meta        | optional named list with metadata.         |

## Details

The canonical AUC format stored in SimiCvizExperiment@auc is always a single data.frame with cells in rows and TFs in columns (the "collected" format), compatible with any GRN method producing per-cell TF activity scores.

**Value**

`SimiCvizExperiment` object.

**Examples**

```
# simic_full <- load_SimiCPipeline(
#   project_dir = "path/to/simic_run",
#   run_name    = "example",
#   lambda1     = "0.01",
#   lambda2     = "0.001")
```

---

`plot_auc_cumulative`      *Plot AUC cumulative (ECDF) curves by label*

---

**Description**

Plot AUC cumulative (ECDF) curves by label

**Usage**

```
plot_auc_cumulative(
  x,
  tf_names = NULL,
  labels = NULL,
  alpha = 0.8,
  rug = FALSE,
  grid = c(4L, 2L),
  percentile = 0.5,
  include_table = TRUE,
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 14,
  height = NULL
)
```

**Arguments**

|                            |                                                             |
|----------------------------|-------------------------------------------------------------|
| <code>x</code>             | A <code>SimiCvizExperiment</code> object.                   |
| <code>tf_names</code>      | Optional TF identifiers to plot. Defaults to all available. |
| <code>labels</code>        | Optional labels to include.                                 |
| <code>alpha</code>         | Line transparency for ECDF curves.                          |
| <code>rug</code>           | Logical; draw rug marks at the x-axis.                      |
| <code>grid</code>          | Integer vector <code>c(nrow, ncol)</code> for page layout.  |
| <code>percentile</code>    | Percentile used in ECDF summary metrics.                    |
| <code>include_table</code> | Logical; add ECDF metrics table below each panel.           |
| <code>save</code>          | Logical; save to PDF.                                       |
| <code>filename</code>      | Output filename when <code>save = TRUE</code> .             |
| <code>out_dir</code>       | Output directory when <code>save = TRUE</code> .            |
| <code>width, height</code> | Output page dimensions.                                     |

**Value**

Invisibly, a list of grobs/plots.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds", package = "SimiCviz"))
plot_auc_cumulative(
  simic,
  tf_names = c("MEF2D", "SATB1"),
  rug = TRUE,
  grid = c(1L, 2L)
)
```

---

plot\_auc\_distributions

*Plot AUC density distributions by label*

---

**Description**

Plot AUC density distributions by label

**Usage**

```
plot_auc_distributions(
  x,
  tf_names = NULL,
  labels = NULL,
  fill = TRUE,
  alpha = 0.5,
  bw_adjust = 1,
  rug = FALSE,
  grid = c(4L, 2L),
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 14,
  height = NULL
)
```

**Arguments**

|           |                                                             |
|-----------|-------------------------------------------------------------|
| x         | A SimiCvizExperiment object.                                |
| tf_names  | Optional TF identifiers to plot. Defaults to all available. |
| labels    | Optional labels to include.                                 |
| fill      | Logical; if TRUE, draw filled densities.                    |
| alpha     | Density transparency.                                       |
| bw_adjust | Bandwidth adjustment passed to density estimation.          |
| rug       | Logical; draw rug marks.                                    |

|               |                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------|
| grid          | Integer vector <code>c(nrow, ncol)</code> for page layout. Use <code>NULL</code> for a single auto-sized page. |
| save          | Logical; save to PDF.                                                                                          |
| filename      | Output filename when <code>save = TRUE</code> .                                                                |
| out_dir       | Output directory when <code>save = TRUE</code> .                                                               |
| width, height | Output page dimensions.                                                                                        |

### Value

Invisibly, a list of ggplot objects.

### Examples

```
simic <- readRDS(system.file("extdata", "simic_full.rds", package = "SimiCviz"))
plot_auc_distributions(
  simic,
  tf_names = c("MEF2D", "SATB1"),
  fill      = TRUE,
  alpha     = 0.6,
  bw_adjust = 1/8,
  rug       = TRUE,
  grid      = c(1L, 2L)
)
```

---

|                  |                                              |
|------------------|----------------------------------------------|
| plot_auc_heatmap | <i>Plot mean AUC heatmap by TF and label</i> |
|------------------|----------------------------------------------|

---

### Description

Plot mean AUC heatmap by TF and label

### Usage

```
plot_auc_heatmap(
  x,
  tf_names = NULL,
  labels = NULL,
  top_n = NULL,
  cmap = "cividis",
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 10,
  height = NULL
)
```

**Arguments**

|               |                                                              |
|---------------|--------------------------------------------------------------|
| x             | A SimiCvizExperiment object.                                 |
| tf_names      | Optional TF identifiers to plot. Defaults to all available.  |
| labels        | Optional labels to include.                                  |
| top_n         | Optional number of TFs to keep by dynamic range.             |
| cmap          | Colour map specification passed to internal palette helpers. |
| save          | Logical; save to PDF.                                        |
| filename      | Output filename when save = TRUE.                            |
| out_dir       | Output directory when save = TRUE.                           |
| width, height | Output page dimensions.                                      |

**Value**

Invisibly, the ggplot object.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds", package = "SimiCviz"))
plot_auc_heatmap(simic, top_n = 5)
```

---

```
plot_auc_summary_statistics
```

*Plot AUC summary statistics*

---

**Description**

Plot AUC summary statistics

**Usage**

```
plot_auc_summary_statistics(
  x,
  labels = NULL,
  high_threshold = 0.5,
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 14,
  height = 10
)
```

**Arguments**

|                |                                            |
|----------------|--------------------------------------------|
| x              | A SimiCvizExperiment object.               |
| labels         | Optional labels to include.                |
| high_threshold | Threshold used to count highly active TFs. |
| save           | Logical; save to PDF.                      |
| filename       | Output filename when save = TRUE.          |
| out_dir        | Output directory when save = TRUE.         |
| width, height  | Output page dimensions.                    |

**Value**

Invisibly, a named list of plot objects.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds", package = "SimiCviz"))
plot_auc_summary_statistics(simic)
```

---

```
plot_dissimilarity_heatmap
      Plot dissimilarity heatmap
```

---

**Description**

Displays a heatmap of TF regulatory dissimilarity scores, optionally broken down by cell group. Uses **ggplot2** tile geometry.

**Usage**

```
plot_dissimilarity_heatmap(
  x,
  top_n = NULL,
  sort_by = NULL,
  dissim_df = NULL,
  cmap = NULL,
  show_values = TRUE,
  save = FALSE,
  out_dir = getwd(),
  filename = NULL,
  width = NULL,
  height = NULL,
  ...
)
```

**Arguments**

|             |                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x           | A SimiCvizExperiment object.                                                                                                                                                                                                                                                                                                                                                                                 |
| top_n       | Integer; number of top TFs to display (default: all).                                                                                                                                                                                                                                                                                                                                                        |
| sort_by     | Column name to sort TFs by. Default: "MinMax_score" (no groups) or "mean_score" (groups).                                                                                                                                                                                                                                                                                                                    |
| dissim_df   | Optional pre-computed dissimilarity data.frame. If NULL, <a href="#">calculate_dissimilarity</a> is called internally.                                                                                                                                                                                                                                                                                       |
| cmap        | Colour palette specification. Can be: <ul style="list-style-type: none"> <li>• A viridis palette name: "viridis", "magma", "plasma", "inferno", or "cividis" (requires <b>viridisLite</b>).</li> <li>• A single colour string (gradient from white to that colour).</li> <li>• A character vector of 2+ colours for a custom gradient.</li> <li>• NULL (default): built-in viridis-like gradient.</li> </ul> |
| show_values | Logical; annotate cells with numeric values (default TRUE).                                                                                                                                                                                                                                                                                                                                                  |

|               |                                                            |
|---------------|------------------------------------------------------------|
| save          | Logical; save the plot to a PDF file (default TRUE).       |
| out_dir       | Output directory for the PDF (default: working directory). |
| filename      | Custom filename (default: auto-generated).                 |
| width, height | PDF dimensions in inches.                                  |
| ...           | Additional arguments passed to calculate_dissimilarity.    |

### Value

Invisibly, a list with plot (the ggplot object) and data (the dissimilarity data.frame).

### Examples

```
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
plot_dissimilarity_heatmap(simic, top_n = 20)
plot_dissimilarity_heatmap(simic, top_n = 5, cmap = "magma")
plot_dissimilarity_heatmap(simic, cmap = c("white", "red", "darkred"),
                           show_values = FALSE)
```

---

plot\_r2\_distribution    *Plot adjusted R<sup>2</sup> distributions*

---

### Description

Plots histograms of adjusted R<sup>2</sup> values per label, similar to the Python SimiCVisualization\$plot\_r2\_distribution.

### Usage

```
plot_r2_distribution(
  adjusted_r_squared,
  x = NULL,
  labels = NULL,
  threshold = 0.7,
  grid = NULL,
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 10,
  height = NULL
)
```

### Arguments

adjusted\_r\_squared

A **named list** of numeric vectors, one per label. Names must be label identifiers (e.g. "0", "1", ...). This metric is specific to SimiC's regression step; other methods may not produce it, so it is kept as an explicit input rather than extracted from the experiment object.

|               |                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x             | Optional SimiCvizExperiment object used solely to resolve display names and colors for labels. If NULL, default "Label i" naming and a built-in palette are used. |
| labels        | Optional vector of labels to plot (subset of names(adjusted_r_squared)). Defaults to all.                                                                         |
| threshold     | Numeric R <sup>2</sup> threshold line and summary-statistic cutoff (default 0.7).                                                                                 |
| grid          | Optional numeric vector of length 2: c(nrow, ncol). If NULL, one label per row is used.                                                                           |
| save          | logical; save to PDF (default FALSE).                                                                                                                             |
| filename      | PDF filename (default "R2_distributions.pdf").                                                                                                                    |
| out_dir       | output directory (default getwd()).                                                                                                                               |
| width, height | page dimensions in inches (defaults 10 x 5*nrow).                                                                                                                 |

### Value

Called for side effects (plots). Returns invisible(NULL).

### Examples

```
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
plot_r2_distribution(simic@meta$adjusted_r_squared, simic,
                    grid = c(2, 2))
plot_r2_distribution(simic@meta$adjusted_r_squared, simic,
                    threshold = 0.9, labels = c(0, 1))
```

---

|                     |                                              |
|---------------------|----------------------------------------------|
| plot_target_weights | <i>Plot weight barplots for target genes</i> |
|---------------------|----------------------------------------------|

---

### Description

For each target gene, shows a grouped bar chart of incoming regulatory weights from all TFs, coloured by phenotype label. Mirrors SimiCVisualization.plot\_target\_weights from the Python package.

### Usage

```
plot_target_weights(
  x,
  target_names = NULL,
  labels = NULL,
  grid = c(4L, 1L),
  top_n = NULL,
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 12,
  height = NULL
)
```

**Arguments**

|               |                                                                                      |
|---------------|--------------------------------------------------------------------------------------|
| x             | SimiCvizExperiment                                                                   |
| target_names  | character vector of targets to plot (default: all)                                   |
| labels        | integer or character vector of labels (default: all)                                 |
| grid          | integer vector c(nrow, ncol) per page (default c(4, 1)). Set NULL for a single page. |
| top_n         | Optional integer. Currently accepted for API compatibility; ignored in target mode.  |
| save          | logical; save to PDF (default FALSE)                                                 |
| filename      | PDF filename (default "target_weights.pdf")                                          |
| out_dir       | output directory (default getwd())                                                   |
| width, height | page dimensions in inches                                                            |

**Value**

Invisibly, a list of page grobs.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
plot_target_weights(simic,
                    target_names = c("CD79A", "IGHM"),
                    labels = c(0, 1), grid = c(2, 1))
```

---

plot\_tf\_network\_heatmap

*Plot heatmap of a TF regulatory network across phenotypes*

---

**Description**

Displays a heatmap showing the regulatory weights of a single transcription factor across its top target genes, with one column per phenotype label. Targets that fail the adjusted- $R^2$  filter are shown as grey cells labelled " $< R^2$  threshold", mirroring the Python SimiCVisualization.plot\_tf\_network\_heatmap() method.

**Usage**

```
plot_tf_network_heatmap(
  x,
  tf_name,
  labels = NULL,
  top_n = 10L,
  r2_threshold = NULL,
  cmap = c("blue", "white", "red"),
  show_values = TRUE,
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
```

```

    width = NULL,
    height = NULL
  )

```

## Arguments

|                            |                                                                                                                                                                                                                                                                                           |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>             | A SimiCvizExperiment object.                                                                                                                                                                                                                                                              |
| <code>tf_name</code>       | Character; name of the transcription factor to plot.                                                                                                                                                                                                                                      |
| <code>labels</code>        | Integer or character vector of labels to include (default: all).                                                                                                                                                                                                                          |
| <code>top_n</code>         | Integer; number of top target genes to display, ranked by maximum absolute weight across labels (default: 10).                                                                                                                                                                            |
| <code>r2_threshold</code>  | Numeric or NULL. Targets with adjusted $R^2$ below this value receive NA and are rendered as grey tiles. Passed to <code>.get_tf_network()</code> .                                                                                                                                       |
| <code>cmap</code>          | Colour palette specification, identical to <a href="#">plot_dissimilarity_heatmap</a> . Recommended diverging palettes such as "RdBu_r" (red-white-blue, reversed) or a two-colour vector like <code>c("blue", "white", "red")</code> . Default: <code>c("blue", "white", "red")</code> . |
| <code>show_values</code>   | Logical; annotate each cell with its weight or "< $R^2$ threshold" text (default TRUE).                                                                                                                                                                                                   |
| <code>save</code>          | Logical; save the plot to a PDF file (default TRUE).                                                                                                                                                                                                                                      |
| <code>filename</code>      | Custom filename (default: auto-generated from <code>tf_name</code> ).                                                                                                                                                                                                                     |
| <code>out_dir</code>       | Output directory for the PDF (default: working directory).                                                                                                                                                                                                                                |
| <code>width, height</code> | PDF dimensions in inches (auto-calculated if NULL).                                                                                                                                                                                                                                       |

## Value

Invisibly, a list with components:

**plot** The ggplot object.

**data** The (filtered) data.frame of weights used for the plot.

## Examples

```

simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
plot_tf_network_heatmap(simic, "SATB1")

# Custom palette and top targets
plot_tf_network_heatmap(simic, "E2F4", top_n = 15,
                        cmap = c("white", "red", "darkred"),
                        r2_threshold = 0.7)

# Only specific labels, no saving
plot_tf_network_heatmap(simic, "MEF2D", labels = c(0, 2),
                        show_values = FALSE)

```

---

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| plot_tf_weights | <i>Plot weight barplots for transcription factors</i> |
|-----------------|-------------------------------------------------------|

---

## Description

For each TF, shows a grouped bar chart of regulatory weights across all target genes, coloured by phenotype label. Mirrors SimiCVisualization.plot\_tf\_weights from the Python package.

## Usage

```
plot_tf_weights(
  x,
  tf_names = NULL,
  labels = NULL,
  top_n = 50L,
  allowed_targets = NULL,
  grid = c(4L, 1L),
  save = FALSE,
  filename = NULL,
  out_dir = getwd(),
  width = 16,
  height = NULL
)
```

## Arguments

|                 |                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x               | SimiCvizExperiment                                                                                                                                               |
| tf_names        | character vector of TFs to plot (default: all)                                                                                                                   |
| labels          | integer or character vector of labels (default: all)                                                                                                             |
| top_n           | integer; show only the top N targets by mean absolute weight (default 50). Applied after allowed_targets filtering.                                              |
| allowed_targets | character vector or NULL; restrict targets to this pre-filtered set before applying top_n. Useful for passing targets that pass an external significance filter. |
| grid            | integer vector c(nrow, ncol) per page (default c(4, 1)). Set NULL for a single page.                                                                             |
| save            | logical; save to PDF (default FALSE)                                                                                                                             |
| filename        | PDF filename (default "TF_weights.pdf")                                                                                                                          |
| out_dir         | output directory (default getwd())                                                                                                                               |
| width, height   | page dimensions in inches                                                                                                                                        |

## Value

Invisibly, a list of page grobs.

**Examples**

```
simic <- readRDS(system.file("extdata", "simic_full.rds",
                             package = "SimiCviz"))
plot_tf_weights(simic, tf_names = c("MEF2D", "SATB1"),
                 top_n = 20, grid = c(2, 1))
```

---

|              |                                                              |
|--------------|--------------------------------------------------------------|
| read_auc_csv | <i>Read SimiC-style AUC results from CSV (compatibility)</i> |
|--------------|--------------------------------------------------------------|

---

**Description**

Read SimiC-style AUC results from CSV (compatibility)

**Usage**

```
read_auc_csv(file, ...)
```

**Arguments**

|      |                                                                                                   |
|------|---------------------------------------------------------------------------------------------------|
| file | path to a CSV file containing AUC metrics. Expected columns include at least cell, tf, and score. |
| ...  | additional arguments passed to <code>utils::read.csv()</code> .                                   |

**Value**

data.frame with AUC metrics.

**Examples**

```
auc_path <- system.file("extdata", "example_auc.csv",
                        package = "SimiCviz")
auc_df <- read_auc_csv(auc_path)
```

---

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| read_auc_pickle | <i>Read SimiCPipeline output AUC matrices (pickle file)</i> |
|-----------------|-------------------------------------------------------------|

---

**Description**

Returns a list of matrices (one per label), each with cells in rows and TFs in columns.

**Usage**

```
read_auc_pickle(file)
```

**Arguments**

|      |                                                          |
|------|----------------------------------------------------------|
| file | path to a pickle file containing per-label AUC matrices. |
|------|----------------------------------------------------------|

**Value**

A named list of data.frames / matrices (one per label).

Examples

```
# This file does not exist in the package due to size constrains, but if you have a SimiCPipeline output, you can

# auc_file <- system.file("extdata",paste0("outputSimic/example_simic_wAUC.pickle"))
# auc_list <- read_auc_pickle(auc_file)
```

---

|             |                          |
|-------------|--------------------------|
| read_pickle | <i>Read pickle files</i> |
|-------------|--------------------------|

---

Description

Read pickle files

Usage

```
read_pickle(file)
```

Arguments

file                      path to the pickle file

Value

object containing the contents of the pickle file

Examples

```
weights_file <- system.file("extdata",
                             file.path("outputSimic/example_simic_weights.pickle"),
                             package = "SimiCviz")
simic_weights <- read_pickle(weights_file)
```

---

|                  |                                                                   |
|------------------|-------------------------------------------------------------------|
| read_weights_csv | <i>Read Long-style TF-target weights from CSV (compatibility)</i> |
|------------------|-------------------------------------------------------------------|

---

Description

Read Long-style TF-target weights from CSV (compatibility)

Usage

```
read_weights_csv(file, ...)
```

Arguments

file                      path to a CSV file containing TF-target weights. Expected minimal columns: tf, target, weight. Additional columns such as label, etc. are preserved.

...                        additional arguments passed to `utils::read.csv()`.

**Value**

data.frame with weights.

**Examples**

```
weight_path <- system.file("extdata", "example_weights.csv",
                           package = "SimiCviz")
weights_df <- read_weights_csv(weight_path)
head(weights_df)
#      tf target    weight label
# 1 MEF2D  RPLP1 -0.7501612      0
# 2  E2F4  RPLP1  0.0000000      0
# 3 SATB1  RPLP1  0.0000000      0
# 4  IRF1  RPLP1 -0.8536968      0
# 5  ATF6  RPLP1 -2.6133661      0
# 6   JUN  RPLP1  0.0000000      0
```

---

|                     |                                                        |
|---------------------|--------------------------------------------------------|
| read_weights_pickle | <i>Read SimiCPipeline output weights (pickle file)</i> |
|---------------------|--------------------------------------------------------|

---

**Description**

Read SimiCPipeline output weights (pickle file)

**Usage**

```
read_weights_pickle(file)
```

**Arguments**

file                      path to the pickle file containing SimiC weights.

**Value**

A list of length n\_labels (phenotypes) with GRN weight matrices (TFs x targets).

**Examples**

```
weights_file <- system.file("extdata",
                           file.path("outputSimic/example_simic_weights.pickle"),
                           package = "SimiCviz")
simic_weights <- read_weights_pickle(weights_file)
```

---

|               |                                            |
|---------------|--------------------------------------------|
| setCellLabels | <i>Set or update cell-to-label mapping</i> |
|---------------|--------------------------------------------|

---

**Description**

Set or update cell-to-label mapping

**Usage**

```
setCellLabels(x, cell_labels)
```

**Arguments**

**x** SimiCvizExperiment.  
**cell\_labels** Any format accepted by [load\\_cell\\_labels](#).

**Value**

Modified SimiCvizExperiment object.

**Examples**

```
wt <- data.frame(tf      = c("TF1", "TF2"),
                 target = c("Gene1", "Gene2"),
                 weight = c(0.5, 0.8),
                 label  = c(0, 1)
                )
simic <- SimiCvizExperiment(weights = wt)

cell_labels <- data.frame(cell = c("Cell1", "Cell2"),
                          label = c(0, 1))
simic <- setCellLabels(simic, cell_labels)
```

---

|               |                                             |
|---------------|---------------------------------------------|
| setLabelNames | <i>Set or update label names and colors</i> |
|---------------|---------------------------------------------|

---

**Description**

Validates that the provided vectors match the number of labels in cell\_labels (preferred) or weights.

**Usage**

```
setLabelNames(x, label_names, colors = NULL)
```

**Arguments**

**x** SimiCvizExperiment.  
**label\_names** named character vector mapping labels to display names.  
**colors** optional named character vector mapping labels to colors.

**Value**

Modified SimiCvizExperiment object.

**Examples**

```
wt <- data.frame(tf      = c("TF1", "TF2"),
                 target = c("Gene1", "Gene2"),
                 weight  = c(0.5, 0.8),
                 label   = c(0, 1)
                )
cell_labels <- data.frame(cell = c("Cell1", "Cell2"),
                          label = c(0, 1)
                         )
simic <- SimiCvizExperiment(weights = wt, cell_labels = cell_labels)
simic <- setLabelNames(simic,
                       label_names = c("control", "treated"),
                       colors = c("#e0e0e0", "#a8c8ff"))
```

---

SimiCviz

*SimiCviz: Visualization and Activity Score analysis of Gene Regulatory Networks*


---

**Description**

SimiCviz provides utilities to:

- Read SimiC or SimiCPipeline output tables from CSV and pickle files.
- Organize weights, AUCs, and metadata into a coherent object.
- Visualize regulatory networks and performance metrics.
- Compute and visualize regulatory dissimilarity across phenotypes (see [calculate\\_dissimilarity](#) and [plot\\_dissimilarity\\_heatmap](#)).
- Export plots (PDF) and processed data tables (CSV) into a reproducible directory hierarchy.

**Details**

The package is designed to be lightweight and focused on visualization, leaving modeling and inference to SimiCPipeline itself.

**Author(s)**

**Maintainer:** Irene Marín-Goñi <imarin.4@alumni.unav.es> ([ORCID](#))

**See Also**

Useful links:

- <https://github.com/ML4BM-Lab/SimiCviz>
- Report bugs at <https://github.com/ML4BM-Lab/SimiCviz/issues>

---

|                    |                                              |
|--------------------|----------------------------------------------|
| SimiCvizExperiment | <i>Construct a SimiCvizExperiment object</i> |
|--------------------|----------------------------------------------|

---

## Description

Construct a SimiCvizExperiment object

## Usage

```
SimiCvizExperiment(
  weights = NULL,
  auc = NULL,
  cell_labels = NULL,
  label_names = NULL,
  colors = NULL,
  meta = list()
)
```

## Arguments

|             |                                                                                                                                                                                                                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| weights     | list with weight matrices and adjusted R <sup>2</sup> .                                                                                                                                                                                                                                                                                                    |
| auc         | optional list containing AUC data. The preferred format is <code>list(collected = df)</code> where <code>df</code> is a <code>cells × TF</code> <code>data.frame</code> . A plain <code>data.frame</code> is automatically wrapped into this structure. This canonical format is compatible with any GRN method that produces per-cell TF activity scores. |
| cell_labels | optional <code>data.frame</code> or vector with cell-to-label mapping. <b>If a plain vector is provided</b> (no cell ids), it must be in the same row order as the AUC matrix.                                                                                                                                                                             |
| label_names | optional named character vector mapping labels to display names. Length must match the number of unique labels in <code>cell_labels</code> (if provided) or <code>weights</code> .                                                                                                                                                                         |
| colors      | optional named character vector mapping labels to colors. Same length requirement as <code>label_names</code> .                                                                                                                                                                                                                                            |
| meta        | optional list of additional metadata.                                                                                                                                                                                                                                                                                                                      |

## Value

An object of class "SimiCvizExperiment".

## Examples

```
# Create a minimal SimiCvizExperiment with weights
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
  dimnames = list(paste0("TF", 1:3),
    paste0("Gene", 1:30)))),
  "1" = as.data.frame(matrix(rnorm(90), nrow = 3),
    dimnames = list(paste0("TF", 1:3),
      paste0("Gene", 1:30))))
simic <- SimiCvizExperiment(weights = weights_list)
```

---

SimiCvizExperiment-class

*SimiCvizExperiment class*


---

## Description

Lightweight S4 container for SimiC visualization data.

## Usage

```
## S4 method for signature 'SimiCvizExperiment'
show(object)
```

## Arguments

object                    A [SimiCvizExperiment](#) object.

## Value

An object of class `SimiCvizExperiment` with initialized slots.

## Slots

`weights` list with weight matrices and adjusted  $R^2$  per label.

`auc` list containing the collected AUC data. The canonical element is `$collected`: a data.frame with cells in rows, TFs in columns, and an optional label column. This format is shared across all GRN methods that produce per-cell TF activity scores.

`cell_labels` data.frame with columns `cell` (character) and `label` (integer). Provides the mapping between cell identifiers and phenotype labels. Required for label-specific AUC visualizations.

`tf_ids` character vector of TF identifiers.

`target_ids` character vector of target gene identifiers.

`label_names` named character vector mapping integer labels to display names.

`colors` named character vector mapping integer labels to colors.

`meta` list with arbitrary metadata.

## Examples

```
# Create a SimiCvizExperiment from weights and AUC data
weights_list <- list("0" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30)))),
                    "1" = as.data.frame(matrix(rnorm(90), nrow = 3,
                                                dimnames = list(paste0("TF", 1:3),
                                                                paste0("Gene", 1:30))))))
cell_labels <- data.frame(cell= c("Cell1", "Cell2"), label = c("0", "1"))
simic <- SimiCvizExperiment(weights = weights_list, cell_labels = cell_labels)
```

# Index

## \* internal

- [.make\\_weight\\_barplot](#), [4](#)
  - [.plot\\_weights\\_paginated](#), [4](#)
  - [.subset\\_auc\\_for\\_label](#), [5](#)
  - [.weights\\_to\\_long](#), [6](#)
- [SimiCviz](#), [35](#)
- [.compute\\_cell\\_auc](#), [3](#)
- [.ensure\\_genes\\_x\\_cells\\_format](#), [3](#)
- [.make\\_weight\\_barplot](#), [4](#)
- [.plot\\_weights\\_paginated](#), [4](#)
- [.subset\\_auc\\_for\\_label](#), [5](#)
- [.weights\\_to\\_long](#), [6](#)
- [auc\\_list\\_to\\_df](#), [8](#)
- [AUCProcessor](#), [6](#), [7](#), [13](#), [15](#)
- [AUCProcessor-class](#), [7](#)
- [calculate\\_activity\\_scores](#), [9](#)
- [calculate\\_dissimilarity](#), [10](#), [25](#), [35](#)
- [calculate\\_ecdf\\_auc](#), [11](#)
- [compute\\_auc](#), [12](#)
- [compute\\_auc](#), [AUCProcessor-method](#), [13](#)
- [export\\_SimiCviz\\_csv](#), [13](#)
- [get\\_auc](#), [14](#)
- [get\\_auc](#), [AUCProcessor-method](#), [15](#)
- [get\\_tf\\_network](#), [15](#)
- [is.SimiCvizExperiment](#), [16](#)
- [load\\_cell\\_labels](#), [8](#), [17](#), [34](#)
- [load\\_collected\\_auc](#), [18](#)
- [load\\_expression\\_matrix](#), [18](#)
- [load\\_from\\_csv](#), [19](#)
- [load\\_SimiCPipeline](#), [20](#)
- [plot\\_auc\\_cumulative](#), [21](#)
- [plot\\_auc\\_distributions](#), [22](#)
- [plot\\_auc\\_heatmap](#), [23](#)
- [plot\\_auc\\_summary\\_statistics](#), [24](#)
- [plot\\_dissimilarity\\_heatmap](#), [25](#), [29](#), [35](#)
- [plot\\_r2\\_distribution](#), [26](#)
- [plot\\_target\\_weights](#), [27](#)
- [plot\\_tf\\_network\\_heatmap](#), [28](#)

- [plot\\_tf\\_weights](#), [30](#)
- [read.csv](#), [18](#)
- [read\\_auc\\_csv](#), [31](#)
- [read\\_auc\\_pickle](#), [31](#)
- [read\\_pickle](#), [32](#)
- [read\\_weights\\_csv](#), [32](#)
- [read\\_weights\\_pickle](#), [33](#)
- [setCellLabels](#), [34](#)
- [setLabelNames](#), [34](#)
- [show](#), [AUCProcessor-method](#)
  - [\(AUCProcessor-class\)](#), [7](#)
- [show](#), [SimiCvizExperiment-method](#)
  - [\(SimiCvizExperiment-class\)](#), [37](#)
- [SimiCviz](#), [35](#)
- [SimiCviz-package \(SimiCviz\)](#), [35](#)
- [SimiCvizExperiment](#), [14](#), [19](#), [21](#), [36](#), [37](#)
- [SimiCvizExperiment-class](#), [37](#)
- [utils::read.csv\(\)](#), [31](#), [32](#)