

Package ‘scDiffCom’

October 14, 2025

Type Package

Title Differential Analysis of Intercellular Communication from
scRNA-Seq Data

Version 1.2.0

Description Analysis tools to investigate changes in intercellular communication from scRNA-seq data. Using a Seurat object as input, the package infers which cell-cell interactions are present in the dataset and how these interactions change between two conditions of interest (e.g. young vs old). It relies on an internal database of ligand-receptor interactions (available for human, mouse and rat) that have been gathered from several published studies. Detection and differential analyses rely on permutation tests. The package also contains several tools to perform over-representation analysis and visualize the results. See Lager, C. et al. (2023) <[doi:10.1038/s43587-023-00514-x](https://doi.org/10.1038/s43587-023-00514-x)> for a full description of the methodology.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

RoxygenNote 7.3.3

Suggests biomaRt, covr, DT, ggplot2, GOSemSim, igraph, kableExtra, KEGGREST, knitr, ontologyIndex, ontoProc, pkgdown, plotly, RColorBrewer, rmarkdown, rrvgo, spelling, shiny, shinythemes, shinyWidgets, testthat (>= 3.0.0), visNetwork

Imports data.table, DelayedArray, future, future.apply, lifecycle, magrittr, methods, Seurat (>= 5.0.0), stats, utils

Collate 'data.R' 'interaction_analysis.R' 'objects.R' 'utils_LRI.R'
'utils_cci.R' 'utils_filtering.R' 'utils_interactiveNetworks.R'
'utils_ora.R' 'utils_permutation.R' 'utils_preprocessing.R'
'utils_validation.R' 'utils_shiny.R'

URL <https://cyrillager.github.io/scDiffCom/>,
<https://github.com/CyrillLager/scDiffCom>

Language en-US
Config/testthat/edition 3
BugReports <https://github.com/CyrilLagger/scDiffCom/issues>
NeedsCompilation no
Author Cyril Lagger [aut, cre] (ORCID:
 <https://orcid.org/0000-0003-1701-6896>),
 Eugen Ursu [aut],
 Anais Equey [ctb]
Maintainer Cyril Lagger <lagger.cyril@gmail.com>
Repository CRAN
Date/Publication 2025-10-14 15:10:02 UTC

Contents

BuildNetwork	2
BuildShiny	3
EraseRawCCI	4
FilterCCI	5
gene_ontology_level	6
GetDistributions	7
GetParameters	7
GetTableCCI	8
GetTableORA	8
LRI_human	9
LRI_mouse	10
LRI_rat	11
PlotORA	12
ReduceGO	13
RunORA	14
run_interaction_analysis	15
scDiffCom-class	18
seurat_sample_tms_liver	19
show,scDiffCom-method	20
Index	21

BuildNetwork	<i>Display cell-type to cell-type interactive networks</i>
--------------	--

Description

Create and plot an interactive network that summarize how cell-types and their interactions are over-represented.

Usage

```
BuildNetwork(
  object,
  network_type = c("ORA_network"),
  layout_type = c("bipartite", "conventional"),
  abbreviation_table = NULL
)

## S4 method for signature 'scDiffCom'
BuildNetwork(
  object,
  network_type = c("ORA_network"),
  layout_type = c("bipartite", "conventional"),
  abbreviation_table = NULL
)
```

Arguments

object	scDiffCom object
network_type	Type of network to display. Currently, only ORA_network (default) is supported.
layout_type	Layout of the network to display. Can either be "bipartite" (default) or "conventional".
abbreviation_table	Table with abbreviations for the cell types present in the object. If NULL (default), full names of the cell-types are displayed. Otherwise, it must be a data.frame or data.table with exactly two columns with names ORIGINAL_CELLTYPE and ABBR_CELLTYPE.

Value

A visNetwork object.

BuildShiny

A shiny app to display scDiffCom results

Description

Launch a shiny app to explore scDiffCom results

Usage

```
BuildShiny(object, reduced_go_table = NULL, ...)

## S4 method for signature 'scDiffCom'
BuildShiny(object, reduced_go_table = NULL, ...)
```

Arguments

object	scDiffCom object
reduced_go_table	If NULL (default), over-represented GO terms are displayed as dot plots. If the output of <code>scDiffCom::ReduceGO(object)</code> , GO terms are displayed on a on treemap based on their semantic similarity and over-representation score
...	Additional parameters to <code>shiny::runApp</code>

Value

Launch a shiny app

EraseRawCCI

Create a copy of a scDiffCom object without cci_table_raw

Description

This function will replace `cci_table_raw` by an empty list. Useful to save space for large datasets. However, after this operation, no filtering can be re-run on the new object, meaning that obtaining results for different filtering parameters will require the perform the full analysis from scratch.

Usage

```
EraseRawCCI(object)

## S4 method for signature 'scDiffCom'
EraseRawCCI(object)
```

Arguments

object	scDiffCom object
--------	------------------

Value

A scDiffCom object with an empty list for `cci_table_raw`.

FilterCCI

*Filter a scDiffCom object with new filtering parameters***Description**

Filtering (and ORA) is performed with new parameter on an existing scDiffCom object. The slots cci_table_detected and ora_table are updated accordingly.

Usage

```
FilterCCI(
  object,
  new_threshold_quantile_score = NULL,
  new_threshold_p_value_specificity = NULL,
  new_threshold_p_value_de = NULL,
  new_threshold_logfc = NULL,
  skip_ora = FALSE,
  extra_annotations = NULL,
  verbose = TRUE
)

## S4 method for signature 'scDiffCom'
FilterCCI(
  object,
  new_threshold_quantile_score = NULL,
  new_threshold_p_value_specificity = NULL,
  new_threshold_p_value_de = NULL,
  new_threshold_logfc = NULL,
  skip_ora = FALSE,
  extra_annotations = NULL,
  verbose = TRUE
)
```

Arguments

object scDiffCom object

new_threshold_quantile_score
New threshold value to update threshold_quantile_score. If NULL (default), the value is not updated.

new_threshold_p_value_specificity
New threshold value to update threshold_p_value_specificity. If NULL (default), the value is not updated.

new_threshold_p_value_de
New threshold value to update threshold_p_value_de. If NULL (default), the value is not updated.

new_threshold_logfc	New threshold value to update threshold_logfc. If NULL (default), the value is not updated.
skip_ora	Default is FALSE. If TRUE, ORA is not performed with the new parameters and ora_table is set to an empty list. May be useful if one wants to quickly test (loop-over) several values of parameters and by-pass the ORA computing time.
extra_annotations	Convenience parameter to perform ORA on user-defined non-standard categories. If NULL (default), ORA is performed on standard categories. Otherwise it must be a list of data.tables or data.frames (see Details).
verbose	If TRUE (default) progress messages are printed.

Details

When FilterCCI is called with new parameters, both cci_table_detected and ora_table are updated. For ORA, a call to RunORA is automatically performed on all standard categories. Additional user-defined ORA categories can be added via the parameter extra_annotations. The data.frames or data.tables in this list must have exactly two columns that indicates a relationship between values from a standard category (first column) to values of the new category (second column). As a typical example, this [vignette](#) shows how to perform ORA on cell type families attached to each cell type.

Value

A scDiffCom object with updated results in cci_table_detected and ora_table.

gene_ontology_level	<i>All gene ontology terms annotated with their levels</i>
---------------------	--

Description

This data.table contains all GO terms retrieved from the package ontoProc. Each term is annotated with its number of ancestors, parents and children, as well as with its level (i.e. depth) in the gene ontology graph. Levels are computed by scDiffCom according to scDiffCom::get_GO_LEVELS().

Usage

```
data(gene_ontology_level)
```

Format

A data.table

References

[ontoProc](#)

GetDistributions	<i>Return the slot distributions from a scDiffCom object</i>
------------------	--

Description

Return the slot distributions from a scDiffCom object

Usage

```
GetDistributions(object)
```

```
## S4 method for signature 'scDiffCom'  
GetDistributions(object)
```

Arguments

object	scDiffCom object
--------	------------------

Value

List of matrices with the null distributions of each CCI.

GetParameters	<i>Return the slot parameters from a scDiffCom object</i>
---------------	---

Description

Return the parameters that have been passed to [run_interaction_analysis](#) as well as a few other parameters computed alongside the analysis.

Usage

```
GetParameters(object)
```

```
## S4 method for signature 'scDiffComBase'  
GetParameters(object)
```

Arguments

object	scDiffCom object
--------	------------------

Value

A list of parameters.

GetTableCCI	<i>Return (a subset) of the slot cci_table_raw or cci_table_detected from a scDiffCom object</i>
-------------	--

Description

Return (a subset) of the slot cci_table_raw or cci_table_detected from a scDiffCom object

Usage

```
GetTableCCI(object, type, simplified)
```

```
## S4 method for signature 'scDiffCom'
```

```
GetTableCCI(object, type = c("detected", "raw"), simplified = TRUE)
```

Arguments

object	scDiffCom object
type	Table to extract information from. Can be either "detected" (default) or "raw".
simplified	If TRUE (default) only the most informative columns of the data.table are returned.

Value

A data.table.

GetTableORA	<i>Return some or all ORA tables from the slot ora_table from a scDiffCom object</i>
-------------	--

Description

Return some or all ORA tables from the slot ora_table from a scDiffCom object

Usage

```
GetTableORA(object, categories, simplified)
```

```
## S4 method for signature 'scDiffCom'
```

```
GetTableORA(object, categories = "all", simplified = TRUE)
```

Arguments

object	scDiffCom object
categories	Names of the ORA categories to return. If "all" (default), returns all of them.
simplified	If TRUE (default) only the most informative columns of the data.table are returned.

Value

A list of data.tables.

LRI_human	<i>A collection of human ligand-receptor interactions.</i>
-----------	--

Description

This dataset contains a data.table of curated human ligand-receptor interactions as well as related annotations (GO Terms, KEGG Pathways) and metadata.

Usage

```
data(LRI_human)
```

Format

A list with the following items:

1. LRI_curated: a data.table of curated LRIs
2. LRI_curated_GO: a data.table with GO terms attached to curated LRIs
3. LRI_curated_KEGG: a data.table with KEGG pathways attached to curated LRIs
4. LRI_retrieved_dates: dates at which data have been retrieved from the seven external databases
5. LRI_retrieved_from: paths or packages from where data have been retrieved
6. LRI_biomart_ensembl_version: version of ensembl used for GO annotation

Details

The dataset has been built internally in scDiffCom according to `scDiffCom:::build_LRI(species = "human")`. The LRIs have been retrieved from seven databases (see References). Note that only curated LRIs have been kept.

References

CellChat ([PMID: 33597522](#)), CellPhoneDB ([PMID: 32103204](#)), CellTalkDB ([PMID: 33147626](#)), connectomeDB2020 ([PMID: 33024107](#)), ICELLNET ([PMID: 33597528](#)), NicheNet ([PMID: 31819264](#)), SingleCellSignalR ([PMID: 32196115](#))

LRI_mouse

*A collection of mouse ligand-receptor interactions.***Description**

This dataset contains a data.table of curated mouse ligand-receptor interactions as well as related annotations (GO Terms, KEGG Pathways) and metadata.

Usage

```
data(LRI_mouse)
```

Format

A list with the following items:

1. LRI_curated: a data.table of curated LRIs
2. LRI_curated_GO: a data.table with GO terms attached to curated LRI
3. LRI_curated_KEGG: a data.table with KEGG pathways attached to curated LRIs
4. LRI_retrieved_dates: dates at which data have been retrieved from the seven external databases
5. LRI_retrieved_from: paths or packages from where data have been retrieved
6. LRI_biomart_ensembl_version: version of ensembl used for GO annotation and orthology conversion

Details

The dataset has been built internally in scDiffCom according to `scDiffCom:::build_LRI(species = "mouse")`. The LRIs have been retrieved from seven databases (see References). Note that only curated LRIs have been kept.

References

CellChat ([PMID: 33597522](#)), CellPhoneDB ([PMID: 32103204](#)), CellTalkDB ([PMID: 33147626](#)), connectomeDB2020 ([PMID: 33024107](#)), ICELLNET ([PMID: 33597528](#)), NicheNet ([PMID: 31819264](#)), SingleCellSignalR ([PMID: 32196115](#))

LRI_rat*A collection of rat ligand-receptor interactions.*

Description

This dataset contains a data.table of curated rat ligand-receptor interactions as well as related annotations (GO Terms, KEGG Pathways) and metadata.

Usage

```
data(LRI_rat)
```

Format

A list with the following items:

1. LRI_curated: a data.table of curated LRIs
2. LRI_curated_GO: a data.table with GO terms attached to curated LRI
3. LRI_curated_KEGG: a data.table with KEGG pathways attached to curated LRIs
4. LRI_retrieved_dates: dates at which data have been retrieved from the seven external databases
5. LRI_retrieved_from: paths or packages from where data have been retrieved
6. LRI_biomart_ensembl_version: version of ensembl used for GO annotation and orthology conversion

Details

The dataset has been built internally in scDiffCom according to `scDiffCom:::build_LRI(species = "rat")`. The LRIs have been retrieved from seven databases (see References). Note that only curated LRIs have been kept.

References

CellChat ([PMID: 33597522](#)), CellPhoneDB ([PMID: 32103204](#)), CellTalkDB ([PMID: 33147626](#)), connectomeDB2020 ([PMID: 33024107](#)), ICELLNET ([PMID: 33597528](#)), NicheNet ([PMID: 31819264](#)), SingleCellSignalR ([PMID: 32196115](#))

PlotORA

Display top over-represented keywords from a category of interest

Description

Plot a graph that shows the top over-represented terms of a given category for a given regulation. Terms are ordered by their ORA scores, computed from their odds ratios and adjusted p-values.

Usage

```
PlotORA(
  object,
  category,
  regulation = c("UP", "DOWN", "FLAT"),
  max_terms_show = 20,
  GO_aspect = c("biological_process", "molecular_function", "cellular_component"),
  OR_threshold = 1,
  bh_p_value_threshold = 0.05
)

## S4 method for signature 'scDiffCom'
PlotORA(
  object,
  category,
  regulation = c("UP", "DOWN", "FLAT"),
  max_terms_show = 20,
  GO_aspect = c("biological_process", "molecular_function", "cellular_component"),
  OR_threshold = 1,
  bh_p_value_threshold = 0.05
)
```

Arguments

object	scDiffCom object
category	ORA category to display. Must be the name of one of the category present in ora_table.
regulation	ORA regulation to display. Can be either UP (default), DOWN or FLAT.
max_terms_show	Maximum number of terms to display. Default is 20.
GO_aspect	Name of the GO aspect to display when category == "GO_TERMS". Can be either biological_process (default), molecular_function or cellular_component.
OR_threshold	Only the terms with an odds ratio above this threshold will be displayed. Default is 1, meaning no filtering is performed.
bh_p_value_threshold	Only the terms with an adjusted p-value below this threshold (and always below 0.05) will be displayed. Default is 0.05.

Details

The ORA score is computed as the product between $\log_2(\text{odds ratio})$ and $-\log_{10}(\text{adj. p-value})$.

Value

A ggplot object.

ReduceGO	<i>Reduce scDiffCom GO Terms</i>
----------	----------------------------------

Description

Perform semantic similarity analysis and reduction of the overrepresented GO terms of an scDiffCom object.

Usage

```
ReduceGO(
  object,
  method = c("Rel", "Resnik", "Lin", "Jiang", "Wang"),
  threshold = 0.7
)

## S4 method for signature 'scDiffCom'
ReduceGO(
  object,
  method = c("Rel", "Resnik", "Lin", "Jiang", "Wang"),
  threshold = 0.7
)
```

Arguments

object	scDiffCom object
method	A distance method supported by rrvgo and GOSemSim: c("Rel", "Resnik", "Lin", "Jiang", "Wang")
threshold	Similarity threshold used by rrvgo::reduceSimMatrix

Details

This function is basically a wrapper around rrvgo::calculateSimMatrix and rrvgo::reduceSimMatrix.

Value

A data.table of GO terms with their reduction

RunORA

*Run over-representation analysis***Description**

Perform over-representation analysis (ORA) on a `scDiffCom` object, with the possibility to define new categories in addition to the standard ones supported by default.

Usage

```
RunORA(
  object,
  categories = c("LRI", "LIGAND_COMPLEX", "RECEPTOR_COMPLEX", "ER_CELLTYPES",
    "EMITTER_CELLTYPE", "RECEIVER_CELLTYPE", "GO_TERMS", "KEGG_PWS"),
  extra_annotations = NULL,
  overwrite = TRUE,
  verbose = TRUE
)

## S4 method for signature 'scDiffCom'
RunORA(
  object,
  categories = c("LRI", "LIGAND_COMPLEX", "RECEPTOR_COMPLEX", "ER_CELLTYPES",
    "EMITTER_CELLTYPE", "RECEIVER_CELLTYPE", "GO_TERMS", "KEGG_PWS"),
  extra_annotations = NULL,
  overwrite = TRUE,
  verbose = TRUE
)
```

Arguments

<code>object</code>	<code>scDiffCom</code> object
<code>categories</code>	Names of the standard categories on which to perform ORA. Default is all standard categories, namely <code>c("LRI", "LIGAND_COMPLEX", "RECEPTOR_COMPLEX", "ER_CELLTYPES", "EMITTER_CELLTYPE", "RECEIVER_CELLTYPE", "GO_TERMS", "KEGG_PWS")</code>
<code>extra_annotations</code>	Convenience parameter to perform ORA on user-defined non-standard categories. If <code>NULL</code> (default), ORA is performed only on standard categories from <code>categories</code> . Otherwise it must be a list of <code>data.tables</code> or <code>data.frames</code> (see Details).
<code>overwrite</code>	If <code>TRUE</code> (default), previous results are overwritten in case they correspond to a category passed in <code>categories</code> .
<code>verbose</code>	If <code>TRUE</code> (default), progress messages are printed.

Details

Additional user-defined ORA categories can be added via the parameter `extra_annotatons`. The `data.frames` or `data.tables` in this list must have exactly two columns that indicates a relationship between values from a standard category (first column) to values of the new category (second column). As a typical example, this [vignette](#) shows how to perform ORA on cell type families attached to each cell type.

Value

A `scDiffCom` object with updated slot `ora_table`.

```
run_interaction_analysis
```

Run (differential) intercellular communication analysis

Description

Perform (differential) cell type to cell type communication analysis from a [Seurat object](#), using an internal database of ligand-receptor interactions (LRIs). It infers biologically relevant cell-cell interactions (CCIs) and how they change between two conditions of interest. Over-representation analysis is automatically performed to determine dominant differential signals at the level of the genes, cell types, GO Terms and KEGG Pathways.

Usage

```
run_interaction_analysis(
  seurat_object,
  LRI_species,
  seurat_celltype_id,
  seurat_condition_id,
  iterations = 1000,
  scdiffcom_object_name = "scDiffCom_object",
  seurat_assay = "RNA",
  seurat_layer = "data",
  seurat_slot = deprecated(),
  log_scale = FALSE,
  score_type = "geometric_mean",
  threshold_min_cells = 5,
  threshold_pct = 0.1,
  threshold_quantile_score = 0.2,
  threshold_p_value_specificity = 0.05,
  threshold_p_value_de = 0.05,
  threshold_logfc = log(1.5),
  return_distributions = FALSE,
  seed = 42,
  verbose = TRUE,
  custom_LRI_tables = NULL
)
```

Arguments

seurat_object	Seurat object that must contain normalized data and relevant meta.data columns (see below). Gene names must be MGI (mouse) or HGNC (human) approved symbols.
LRI_species	Either "mouse", "human", "rat" or "custom". Indicates which LRI database to use and corresponds to the species of the seurat_object. Use "custom" at your own risk to use your own LRI table (see custom_LRI_tables).
seurat_celltype_id	Name of the meta.data column in seurat_object that contains cell-type annotations (e.g.: "CELL_TYPE").
seurat_condition_id	List that contains information regarding the two conditions on which to perform differential analysis. Must contain the following three named items: 1. column_name: name of the meta.data column in seurat_object that indicates the condition on each cell (e.g. "AGE") 2. cond1_name: name of the first condition (e.g. "YOUNG") 3. cond2_name: name of the second condition (e.g. "OLD") Can also be set to NULL to only perform a detection analysis (see Details).
iterations	Number of permutations to perform the statistical analysis. The default (1000) is a good compromise for an exploratory analysis and to obtain reasonably accurate p-values in a short time. Otherwise, we recommend using 10000 iterations and to run the analysis in parallel (see Details). Can also be set to 0 for debugging and quickly returning partial results without statistical significance.
scdiffcom_object_name	Name of the scDiffCom S4 object that will be returned ("scDiffCom_object" by default).
seurat_assay	Assay of seurat_object from which to extract data. See Details for an explanation on how data are extracted based on the three parameters seurat_assay, seurat_layer and log_scale.
seurat_layer	Layer of seurat_object from which to extract data. See Details for an explanation on how data are extracted based on the three parameters seurat_assay, seurat_layer and log_scale.
seurat_slot	'r lifecycle::badge("deprecated")' 'seurat_slot' is no longer supported; use 'seurat_layer' instead.
log_scale	When FALSE (the default, recommended), data are treated as normalized but not log1p-transformed. See Details for an explanation on how data are extracted based on the three parameters seurat_assay, seurat_layer and log_scale.
score_type	Metric used to compute cell-cell interaction (CCI) scores. Can either be "geometric_mean" (default) or "arithmetic_mean". It is strongly recommended to use the geometric mean, especially when performing differential analysis. The arithmetic mean might be used when uniquely doing a detection analysis or if the results want to be compared with those of another package.
threshold_min_cells	Minimal number of cells - of a given cell type and condition - required to express a gene for this gene to be considered expressed in the corresponding cell type.

	Incidentally, cell types with less cells than this threshold are removed from the analysis. Set to 5 by default.
threshold_pct	Minimal fraction of cells - of a given cell type and condition - required to express a gene for this gene to be considered expressed in the corresponding cell type. Set to 0.1 by default.
threshold_quantile_score	Threshold value used in conjunction with threshold_p_value_specificity to establish if a CCI is considered "detected". The default (0.2) indicates that CCIs with a score in the 20% lowest-scores are not considered detected. Can be modified without the need to re-perform the permutation analysis (see Details).
threshold_p_value_specificity	Threshold value used in conjunction with threshold_quantile_score to establish if a CCI is considered "detected". CCIs with a (BH-adjusted) specificity p-value above the threshold (0.05 by default) are not considered detected. Can be modified without the need to re-perform the permutation analysis (see Details).
threshold_p_value_de	Threshold value used in conjunction with threshold_logfc to establish how CCIs are differentially expressed between cond1_name and cond2_name. CCIs with a (BH-adjusted) differential p-value above the threshold (0.05 by default) are not considered to change significantly. Can be modified without the need to re-perform the permutation analysis (see Details).
threshold_logfc	Threshold value used in conjunction with threshold_p_value_de to establish how CCIs are differentially expressed between cond1_name and cond2_name. CCIs with an absolute logFC below the threshold ($\log(1.5)$ by default) are considered "FLAT". Can be modified without the need to re-perform the permutation analysis (see Details).
return_distributions	FALSE by default. If TRUE, the distributions obtained from the permutation test are returned alongside the other results. May be used for testing or benchmarking purposes. Can only be enabled when iterations is less than 1000 in order to avoid out of memory issues.
seed	Set a random seed (42 by default) to obtain reproducible results.
verbose	If TRUE (default), print progress messages.
custom_LRI_tables	A list containing a LRI table and, if known, tables with annotations supplied by the user. Overwrite LRI_species and the corresponding internal LRI table. Use to your own risk! Must contain at least the following named item: 1. LRI: a data.table of LRIs The data.table of LRIs must be in the same format as the internal LRI_tables, namely with the columns "LRI", "LIGAND_1", "LIGAND_2", "RECEPTOR_1", "RECEPTOR_2", "RECEPTOR_3". Other named data.tables can be supplied for over-representation analysis (ORA) purposes.

Details

The primary use of this function (and of the package) is to perform differential intercellular communication analysis. However, it is also possible to only perform a detection analysis (by setting `seurat_condition_id` to `NULL`), e.g. if one wants to infer cell-cell interactions from a dataset without having conditions on the cells.

By convention, when performing differential analysis, LOGFC are computed as $\log(\text{score}(\text{cond2_name})/\text{score}(\text{cond1_name}))$. In other words, "UP"-regulated CCIs have a larger score in `cond2_name`.

Parallel computing. If possible, it is recommended to run this function in parallel in order to speed up the analysis for large dataset and/or to obtain better accuracy on the p-values by setting a higher number of iterations. This is as simple as loading the `future` package and setting an appropriate plan (see also our [vignette](#)).

Data extraction. The UMI or read counts matrix is extracted from the assay `seurat_assay` and the layer `seurat_layer`. By default, it is assumed that `seurat_object` contains log1p-transformed normalized data in the layer "data" of its assay "RNA". If `log_scale` is `FALSE` (as recommended), the data are `expm1()` transformed in order to recover normalized values not in log scale.

Modifying filtering parameters (differential analysis only). As long as the slot `cci_table_raw` of the returned `scDiffCom` object is not erased, filtering parameters can be modified to recompute the slots `cci_table_detected` and `ora_table`, without re-performing the time consuming permutation analysis. This may be useful if one wants a fast way to analyze how the results behave in function of, say, different LOGFC thresholds. In practice, this can be done by calling the functions [FilterCCI](#) or [RunORA](#) (see also our [vignette](#)).

Value

An S4 object of class `scDiffCom-class`.

Examples

```
## Not run:
run_interaction_analysis(
  seurat_object = seurat_sample_tms_liver,
  LRI_species = "mouse",
  seurat_celltype_id = "cell_type",
  seurat_condition_id = list(
    column_name = "age_group",
    cond1_name = "YOUNG",
    cond2_name = "OLD"
  )
)

## End(Not run)
```

Description

An object of this class stores the intercellular communication results obtained when calling `run_interaction_analysis`.

Slots

`parameters` List of parameters passed to `run_interaction_analysis` and used to build the object.

`cci_table_raw` `Data.table` with all hypothetical CCIs induced from the original Seurat object and the internal LRI database. Can be erased with `EraseRawCCI` to obtain a lighter object, but might be worth keeping if one intends to modify the filtering parameters (see also our [vignette](#)).

`cci_table_detected` `Data.table` with only the detected CCIs. If `cci_table_raw` is not NULL, can be updated with new filtering parameters without running the full permutation analysis (see `FilterCCI`).

`ora_table` List of `data.tables` with the results of the over-representation analysis for each category. Results for additional categories can be added with `RunORA`.

`distributions` List of matrices with the null distributions of each CCI. NULL by default.

`seurat_sample_tms_liver`

A down-sampled Seurat object to use for testing and benchmarking

Description

This Seurat object has been down-sampled from the original Tabula Muris Senis liver object. Pre-processing and normalization has been performed before down-sampling. It contains 726 features (genes) and 468 samples (cells). It is only intended to be used for testing and benchmarking and does not contain meaningful biological information.

Usage

```
data(seurat_sample_tms_liver)
```

Format

An object of class Seurat.

References

A single-cell transcriptomic atlas characterizes ageing tissues in the mouse, Tabula Muris Consortium (2020) ([PMID: 32669714](#))

show,scDiffCom-method *Display a scDiffCom object*

Description

Display a scDiffCom object

Usage

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

Arguments

object scDiffCom object

Value

Print summary to the console, no return value.

Index

* datasets

- gene_ontology_level, [6](#)
- LRI_human, [9](#)
- LRI_mouse, [10](#)
- LRI_rat, [11](#)
- seurat_sample_tms_liver, [19](#)

BuildNetwork, [2](#)

BuildNetwork, scDiffCom-method
(BuildNetwork), [2](#)

BuildShiny, [3](#)

BuildShiny, scDiffCom-method
(BuildShiny), [3](#)

EraseRawCCI, [4](#), [19](#)

EraseRawCCI, scDiffCom-method
(EraseRawCCI), [4](#)

FilterCCI, [5](#), [18](#), [19](#)

FilterCCI, scDiffCom-method (FilterCCI),
[5](#)

gene_ontology_level, [6](#)

GetDistributions, [7](#)

GetDistributions, scDiffCom-method
(GetDistributions), [7](#)

GetParameters, [7](#)

GetParameters, scDiffComBase-method
(GetParameters), [7](#)

GetTableCCI, [8](#)

GetTableCCI, scDiffCom-method
(GetTableCCI), [8](#)

GetTableORA, [8](#)

GetTableORA, scDiffCom-method
(GetTableORA), [8](#)

LRI_human, [9](#)

LRI_mouse, [10](#)

LRI_rat, [11](#)

PlotORA, [12](#)

PlotORA, scDiffCom-method (PlotORA), [12](#)

ReduceGO, [13](#)

ReduceGO, scDiffCom-method (ReduceGO), [13](#)

run_interaction_analysis, [7](#), [15](#), [19](#)

RunORA, [14](#), [18](#), [19](#)

RunORA, scDiffCom-method (RunORA), [14](#)

scDiffCom-class, [18](#)

seurat_sample_tms_liver, [19](#)

show, scDiffCom-method, [20](#)