Package: CellChat (via r-universe)

May 23, 2026

Type Package

Title Inference and analysis of cell-cell communication from single-cell and spatially resolved transcriptomics data

Version 2.2.0.9001

Author Suoqin Jin

Maintainer Suoqin Jin <sqjin@whu.edu.cn>

Description an open source R tool that infers, visualizes and analyzes the cell-cell communication networks from scRNA-seq and spatially resolved transcriptomics data.

Depends R (>= 3.6.0),dplyr,igraph,ggplot2

License GPL-3

Encoding UTF-8

LazyData TRUE

Imports future, future.apply, pbapply, irlba, NMF (>= 0.23.0), ggalluvial, stringr, svglite, Matrix, ggrepel, circlize (>= 0.4.12), RColorBrewer, cowplot, methods, ComplexHeatmap, RSpectra, Rcpp, reticulate, scales, sna, reshape2, FNN, shape, BiocGenerics, magrittr, patchwork, colorspace, plyr, ggpubr, ggnetwork, BiocNeighbors, plotly,shiny,bslib, collapse

LinkingTo Rcpp, RcppEigen

Suggests rmarkdown, knitr, roxygen2, Seurat (>= 4.0.0), SingleCellExperiment, SummarizedExperiment, purrr,uwot,gg.gap,wordcloud,bsicons,presto,tidyverse,tidyr,rlang

RoxygenNote 7.3.1

VignetteBuilder knitr

Config/pak/sysreqs
cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libglpk-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev perl python3 zlib1g-dev

Repository https://blaserlab.r-universe.dev

Date/Publication 2026-03-04 12:08:33 UTC

RemoteUrl https://github.com/jinworks/CellChat

RemoteRef HEAD

RemoteSha 75253cd0c9e68410e6e721a6d3a0419a1d7e358f

Contents

addMeta *Add the cell information into meta slot*

Description

Add the cell information into meta slot

Usage

```
addMeta(object, meta, meta.name = NULL)
```

Arguments

object	CellChat object
meta	cell information to be added
meta.name	the name of column to be assigned

addReduction *Add a reduced space of the data into CellChat object*

Description

Add a reduced space of the data into CellChat object

Usage

```
addReduction(
  object,
  dr = NULL,
  dr.name = NULL,
  seu.obj = NULL,
  dr.use = NULL,
  force.add = FALSE
)
```

Arguments

object	CellChat object from a single dataset
dr	A data frame (rows are cells with rownames) consisting of a low-dimensional space for visualization
dr.name	A char name of the reduction method for the input ‘dr‘
seu.obj	A Seurat object with the reduced space of the data
dr.use	A char name of the reduction method to use when taking ‘seu.obj‘ as input. By default, all reduced space in ‘seu.obj‘ will be added in ‘object@dr‘
force.add	Whether to force to add a new reduced space when a reduced space exists in ‘object@dr‘

Examples

```
## Not run:
cellChat <- addReduction(object = cellchat, dr = cell.embeddings, dr.name = "umap")

cellChat <- addReduction(object = cellchat, seu.obj = seu.obj)

## End(Not run)
```

`aggregateNet` *Calculate the aggregated network by counting the number of links or summarizing the communication probability*

Description

Calculate the aggregated network by counting the number of links or summarizing the communication probability

Usage

```
aggregateNet(
  object,
  sources.use = NULL,
  targets.use = NULL,
  signaling = NULL,
  pairLR.use = NULL,
  remove.isolate = TRUE,
  thresh = 0.05,
  return.object = TRUE
)
```

Arguments

`object`	CellChat object
`sources.use, targets.use, signaling, pairLR.use`	Please check the description in function `subsetCommunication`
`remove.isolate`	whether removing the isolate cell groups without any interactions when applying `subsetCommunication`
`thresh`	threshold of the p-value for determining significant interaction
`return.object`	whether return an updated CellChat object

Value

Return an updated CellChat object:

'object@net$count' is a matrix: rows and columns are sources and targets respectively, and elements are the number of interactions between any two cell groups. USER can convert a matrix to a data frame using the function 'reshape2::melt()'

'object@net$weight' is also a matrix containing the interaction weights between any two cell groups

'object@net$sum' is deprecated. Use 'object@net$weight'

`alpha_centrality` *compute alpha centrality*

Description

compute alpha centrality

Usage

```
alpha_centrality(g)
```

Arguments

g	a graph objecct

`AnyMatrix-class` *The CellChat Class*

Description

The CellChat object is created from a single-cell transcriptomic data matrix, Seurat V3 or SingleCellExperiment object. When inputting an data matrix, it takes a digital data matrices as input. Genes should be in rows and cells in columns. rownames and colnames should be included. The class provides functions for data preprocessing, intercellular communication network inference, communication network analysis, and visualization.

Details

Class definitions

barPlot *Bar plot for average gene expression*

Description

Please check barplot_internalfor detailed description of the arguments.

Usage

```
barPlot(
  object,
  features,
  group.by = NULL,
  split.by = NULL,
  color.use = NULL,
  method = c("truncatedMean", "triMean", "median"),
  trim = 0.1,
  assay = "RNA",
  x.lab.rot = FALSE,
  ncol = 1,
  ...
)
```

Arguments

object	seurat object
features	Features to plot (gene expression, metrics)
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); pass 'ident' to group by identity class
split.by	Name of a metadata column to split plot by;
color.use	defining the color for each condition/dataset
method	methods for computing the average gene expression per cell group. By default = "truncatedMean", where a value should be assigned to 'trim;
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed.
assay	Name of assay to use, defaults to the active assay
x.lab.rot	whether do rotation for the x.tick.label
ncol	number of columns to show in the plot
...	Extra parameters passed to barplot_internal

Value

ggplot2 object

barplot_internal *Bar plot for dataframe*

Description

Bar plot for dataframe

Usage

```
barplot_internal(
  df,
  x = "cellType",
  y = "value",
  fill = "condition",
  legend.title = NULL,
  width = 0.6,
  title.name = NULL,
  xlabel = NULL,
  ylabel = NULL,
  color.use = NULL,
  remove.xtick = FALSE,
  stat.add = FALSE,
  stat.method = "wilcox.test",
  percent.y = FALSE,
  label.x = 1.5,
  show.legend = TRUE,
  x.lab.rot = FALSE,
  size.text = 10
)
```

Arguments

df	a dataframe
x	Name of one column to show on the x-axis
y	Name of one column to show on the y-axis
fill	Name of one column to compare the values
legend.title	Name of legend
width	bar width
title.name	Name of the main title
xlabel	Name of x label
ylabel	Name of y label
color.use	defining the color of bar plot;
remove.xtick	whether remove x tick
stat.add	whether adding statistical test

`stat.method`, `label.x`
: parameters for ggpubr::stat_compare_means

`percent.y`
: whether showing y-values as percentage

`show.legend`
: Whether show the legend

`x.lab.rot`
: Whether rorate the xtick labels

`size.text`
: font size

Value

ggplot2 object

`buildSNN` *Build SNN matrix*

Description

Build SNN matrix

Usage

```
buildSNN(data.use, k = 10, k.scale = 10, prune.SNN = 1/15)
```

Arguments

`data.use`
: Features x samples matrix to use to build the SNN

`k`
: Defines k for the k-nearest neighbor algorithm

`k.scale`
: Granularity option for k.param

`prune.SNN`
: Sets the cutoff for acceptable Jaccard distances when computing the neighborhood overlap for the SNN construction.

Value

Returns similarity matrix in sparse matrix format

CellChat-class *The key slots used in the CellChat object are described below.*

Description

The key slots used in the CellChat object are described below.

Slots

`data.raw` raw count data matrix

`data` normalized data matrix for CellChat analysis (Genes should be in rows and cells in columns)

`data.signaling` a subset of normalized matrix only containing signaling genes

`data.scale` scaled data matrix

`data.smooth` smoothed data

`images` a list of information of spatial transcriptomics data

`net` a three-dimensional array P (K×K×N), where K is the number of cell groups and N is the number of ligand-receptor pairs. Each row of P indicates the communication probability originating from the sender cell group to other cell groups.

`netP` a three-dimensional array representing cel-cell communication networks on a signaling pathway level

`DB` ligand-receptor interaction database used in the analysis (a subset of CellChatDB)

`LR` a list of information related with ligand-receptor pairs

`meta` data frame storing the information associated with each cell

`idents` a factor defining the cell identity used for all analysis. It becomes a list for a merged CellChat object

`var.features` A list: one element is a vector consisting of the identified over-expressed signaling genes; one element is a data frame returned from the differential expression analysis

`dr` List of the reduced 2D coordinates, one per method, e.g., umap/tsne/dm

`options` List of miscellaneous data, such as parameters used throughout analysis, and a indicator whether the CellChat object is a single or merged

CellChat_theme_opts *ggplot theme in CellChat*

Description

ggplot theme in CellChat

Usage

```
CellChat_theme_opts()
```

CellChatDB.human *Ligand-receptor interactions in CellChat database for human*

Description

The ligand-receptor interaction database curated in CellChat tool

Usage

```
CellChatDB.human
```

Format

A list includes the ligand-receptor interactions

Source

https://github.com/sqjin/CellChat/

CellChatDB.mouse *Ligand-receptor interactions in CellChat database for mouse*

Description

The ligand-receptor interaction database curated in CellChat tool

Usage

```
CellChatDB.mouse
```

Format

A list includes the ligand-receptor interactions

Source

https://github.com/sqjin/CellChat/

CellChatDB.zebrafish *Ligand-receptor interactions in CellChat database for Zebrafish*

Description

The ligand-receptor interaction database curated in CellChat tool

Usage

```
CellChatDB.zebrafish
```

Format

A list includes the ligand-receptor interactions

Source

https://github.com/sqjin/CellChat/

checkGeneSymbol *check the official Gene Symbol*

Description

check the official Gene Symbol

Usage

```
checkGeneSymbol(geneSet, geneIfo)
```

Arguments

geneSet	gene set to check
geneIfo	official Gene Symbol

colorRamp3 *Color interpolation*

Description

This function is modified from https://rdrr.io/cran/circlize/src/R/utils.R Colors are linearly interpolated according to break values and corresponding colors through CIE Lab color space (‘colorspace::LAB‘) by default. Values exceeding breaks will be assigned with corresponding maximum or minimum colors.

Usage

```
colorRamp3(breaks, colors, transparency = 0, space = "LAB")
```

Arguments

breaks	A vector indicating numeric breaks
colors	A vector of colors which correspond to values in “breaks“
transparency	A single value in “[0, 1]“. 0 refers to no transparency and 1 refers to full transparency
space	color space in which colors are interpolated. Value should be one of "RGB", "HSV", "HLS", "LAB", "XYZ", "sRGB", "LUV", see ‘colorspace::color-class‘ for detail.

Value

It returns a function which accepts a vector of numeric values and returns interpolated colors.

Examples

```
## Not run:
col_fun = colorRamp3(c(-1, 0, 1), c("green", "white", "red"))
col_fun(c(-2, -1, -0.5, 0, 0.5, 1, 2))

## End(Not run)
```

compareInteractions *Comparing the number of inferred communication links between different datasets*

Description

Comparing the number of inferred communication links between different datasets

Usage

```
compareInteractions(
  object,
  measure = c("count", "weight"),
  color.use = NULL,
  group = NULL,
  group.levels = NULL,
  group.facet = NULL,
  group.facet.levels = NULL,
  n.row = 1,
  color.alpha = 1,
  legend.title = NULL,
  width = 0.6,
  title.name = NULL,
  digits = 3,
  xlabel = NULL,
  ylabel = NULL,
  remove.xtick = FALSE,
  show.legend = TRUE,
  x.lab.rot = FALSE,
  angle.x = 45,
  vjust.x = NULL,
  hjust.x = 1,
  size.text = 10
)
```

Arguments

object	A merged CellChat object
measure	"count" or "weight". "count": comparing the number of interactions; "weight": comparing the total interaction weights (strength)
color.use	defining the color for each group of datasets
group	a vector giving the groups of different datasets to define colors of the bar plot. Default: only one group and a single color
group.levels	the factor level in the defined group
group.facet	Name of one metadata column defining faceting groups
group.facet.levels	the factor level in the defined group.facet
n.row	Number of rows in facet_grid()
color.alpha	transparency
legend.title	legend title
width	bar width
title.name	main title of the plot
digits	integer indicating the number of decimal places (round) to be used when ‘measure‘ is ‘weight‘.

xlabel	label of x-axis
ylabel	label of y-axis
remove.xtick	whether remove xtick
show.legend	whether show the legend
x.lab.rot, angle.x, vjust.x, hjust.x	adjusting parameters if rotating xtick.labels when x.lab.rot = TRUE
size.text	font size of the text

Value

A ggplot object

`computeAveExpr` *Compute averaged expression values for each cell group*

Description

Compute averaged expression values for each cell group

Usage

```
computeAveExpr(
  object,
  features = NULL,
  group.by = NULL,
  type = c("triMean", "truncatedMean", "median"),
  trim = NULL,
  slot.name = c("data.signaling", "data"),
  data.use = NULL
)
```

Arguments

object	CellChat object
features	a char vector giving the used features. default use all features
group.by	cell group information; default is ‘object@idents‘ when input is a single object and ‘object@idents$joint‘ when input is a merged object; otherwise it should be one of the column names of the meta slot
type	methods for computing the average gene expression per cell group. By default = "triMean", defined as a weighted average of the distribution's median and its two quartiles (https://en.wikipedia.org/wiki/Trimean); When setting ‘type = "truncatedMean"‘, a value should be assigned to ’trim’. See the function ‘base::mean‘.
trim	the fraction (0 to 0.25) of observations to be trimmed from each end of x before the mean is computed.

slot.name	the data in the slot.name to use
data.use	a customed data matrix. Default: data.use = NULL and the expression matrix in the 'slot.name' is used

Value

Returns a matrix with genes as rows, cell groups as columns.

`computeCellDistance` *Compute cell-cell distance based on the spatial coordinates*

Description

Compute cell-cell distance based on the spatial coordinates

Usage

```
computeCellDistance(
  coordinates,
  interaction.range = NULL,
  ratio = NULL,
  tol = NULL
)
```

Arguments

coordinates	a data matrix in which each row gives the spatial locations/coordinates of each cell/spot
interaction.range	The maximum interaction/diffusion range of ligands. This hard threshold is used to filter out the connections between spatially distant cells
ratio	The conversion factor when converting spatial coordinates from Pixels or other units to Micrometers (i.e.,Microns). For example, setting 'ratio = 0.18' indicates that 1 pixel equals 0.18um in the coordinates. For 10X visium, it is the ratio of the theoretical spot size (i.e., 65um) over the number of pixels that span the diameter of a theoretical spot size in the full-resolution image (i.e., 'spot.size.fullres' in the 'scalefactors_json.json' file).
tol	The tolerance factor to increase the robustness when comparing the center-to-center distance against the 'interaction.range'. This can be the half value of cell/spot size in the unit of um. For example, for 10X visium, 'tol' can be set as '65/2'; for slide-seq, 'tol' can be set as '10/2'. If the cell/spot size is not known, we provide a function 'computeCellDistance' to compute the center-to-center distance. 'tol' can be the the half value of the minimum center-to-center distance.

Value

an object of class "dist" giving the pairwise cell-cell distance

computeCentralityLocal *Compute Centrality measures for a signaling network*

Description

Compute Centrality measures for a signaling network

Usage

```
computeCentralityLocal(net)
```

Arguments

net	compute the centrality measures on a specific signaling network given by a 2 or 3 dimemsional array net

computeCommunProb *Compute the communication probability/strength between any interacting cell groups*

Description

To further speed up on large-scale datasets, USER can downsample the data using the function 'subset' from Seurat package (e.g., pbmc.small <- subset(pbmc, downsample = 500)), or using the function 'sketchData' from CellChat, in particular for the large cell clusters;

Usage

```
computeCommunProb(
  object,
  type = c("triMean", "truncatedMean", "thresholdedMean", "median"),
  trim = 0.1,
  LR.use = NULL,
  raw.use = TRUE,
  population.size = FALSE,
  distance.use = TRUE,
  interaction.range = 250,
  scale.distance = 0.01,
  k.min = 10,
  contact.dependent = TRUE,
  contact.range = NULL,
  contact.knn.k = NULL,
  contact.dependent.forced = FALSE,
  do.symmetric = TRUE,
```

```
  nboot = 100,
  seed.use = 1L,
  Kh = 0.5,
  n = 1
)
```

Arguments

object	CellChat object
type	Methods for computing the average gene expression per cell group. By default = "triMean", producing fewer but stronger interactions; When setting 'type = "truncatedMean"', a value should be assigned to 'trim', producing more interactions.
trim	the fraction (0 to 0.25) of observations to be trimmed from each end of x before the mean is computed
LR.use	A subset of ligand-receptor interactions used in inferring communication network
raw.use	Whether use the raw data (i.e., 'object@data.signaling') or the smoothed data (i.e., 'object@data.smooth'). Set raw.use = FALSE to use the projected data when analyzing single-cell data with shallow sequencing depth because the projected data could help to reduce the dropout effects of signaling genes, in particular for possible zero expression of subunits of ligands/receptors.
population.size	Whether consider the proportion of cells in each group across all sequenced cells. Set population.size = FALSE if analyzing sorting-enriched single cells, to remove the potential artifact of population size. Set population.size = TRUE if analyzing unsorted single-cell transcriptomes, with the reason that abundant cell populations tend to send collectively stronger signals than the rare cell populations.
	Parameters for spatial data analysis:
distance.use	Whether to use distance constraints to compute communication probability. Setting 'distance.use = TRUE' indicates that the cell-cell communication probability is inversely proportional to the computed distance. Setting 'distance.use = FALSE' will only filter out interactions between spatially distant regions, but not add distance constraints.
interaction.range	The maximum interaction/diffusion length of ligands (Unit: microns). This hard threshold is used to filter out the connections between spatially distant regions
scale.distance	A scale or normalization factor for the spatial distances when setting 'distance.use = TRUE'. For example, scale.distance equals 1, 0.1, 0.01, 0.001, 0.11, or 0.011. We choose this values such that the minimum value of the scaled distances is in [1,2]. This value is not necessary when setting 'distance.use = FALSE'.
	When comparing communication across different CellChat objects, the same scale factor should be used. For a single CellChat analysis, different scale factors will not affect the ranking of the signaling based on their interaction strength.
k.min	The minimum number of interacting cell pairs required for defining spatially proximal cell groups.

`contact.dependent`
: Whether using the ‘contact-dependent‘ manner for inference signaling, that is determining interacting cell pairs by requiring cells to be in direct membrane-membrane contact. By default ‘contact.dependent = TRUE‘ when inferring contact-dependent and juxtacrine signaling (that is "Cell-Cell Contact" signaling classified in CellChatDB$interaction$annotation). If only focusing on ‘Secreted Signaling‘, the ‘contact-dependent‘ manner will be not used except for setting ‘contact.dependent.forced = TRUE‘.

`contact.range`
: The interaction range (Unit: microns) to restrict the contact-dependent signaling when ‘contact.dependent = TRUE‘. For spatial transcriptomics in a single-cell resolution, ‘contact.range‘ is approximately equal to the estimated cell diameter (i.e., the cell center-to-center distance), which means that contact-dependent and juxtacrine signaling can only happens when the two cells are contact to each other.

  Typically, ‘contact.range = 10‘, which is a typical human cell size. However, for low-resolution spatial data such as 10X visium, it should be the cell center-to-center distance (i.e., ‘contact.range = 100‘ for visium data). The function ‘computeCellDistance‘ can compute the center-to-center distance.

`contact.knn.k`
: Number of neighbors to restrict the contact-dependent signaling within the neatest neighbors when ‘contact.dependent = TRUE‘. By default, CellChat uses ‘contact.range‘ to restrict the contact-dependent signaling; however, users can also provide a value of ‘contact.knn.k‘, in order to determine interacting cell pairs based on the k-nearest neighbors (knn). For 10X visium, contact.knn.k = 6. For other spatial technologies, this value may be hard to determine because the sequenced cells/spots are usually not regularly arranged.

`contact.dependent.forced`
: Whether forcing to use the ‘contact-dependent‘ manner for inference signaling for all L-R pairs including secreted signaling. Users can set ‘contact.dependent.forced = TRUE‘ if also preferring interactions within a contact manner for ‘Secreted Signaling‘.

`do.symmetric`
: Whether converting the adjacent matrix into symmetric one when determining spatially proximal cell groups. Default is TRUE, indicating that if adj(i,j) or adj(j,i) is zero, then both are zeros.

`nboot`
: Threshold of p-values

`seed.use`
: Set a random seed. By default, set the seed to 1.

`Kh`
: Parameter in Hill function

`n`
: Parameter in Hill function

Value

A CellChat object with updated slot ’net’:

object@net$prob is the inferred communication probability (strength) array, where the first, second and third dimensions represent a source, target and ligand-receptor pair, respectively.

USER can access all the inferred cell-cell communications using the function ’subsetCommunication(object)’, which returns a data frame.

object@net$pval is the corresponding p-values of each interaction

`computeCommunProbPathway` *Compute the communication probability on signaling pathway level by summarizing all related ligands/receptors*

Description

Compute the communication probability on signaling pathway level by summarizing all related ligands/receptors

Usage

```
computeCommunProbPathway(
  object = NULL,
  net = NULL,
  pairLR.use = NULL,
  thresh = 0.05
)
```

Arguments

`object`	CellChat object
`net`	A list from object@net; If net = NULL, net = object@net
`pairLR.use`	A dataframe giving the ligand-receptor interactions; If pairLR.use = NULL, pairLR.use = object@LR$LRsig
`thresh`	threshold of the p-value for determining significant interaction

Value

A CellChat object with updated slot 'netP':

object@netP$prob is the communication probability array on signaling pathway level; USER can convert this array to a data frame using the function 'reshape2::melt()',

e.g., 'df.netP <- reshape2::melt(object@netP$prob, value.name = "prob"); colnames(df.netP)[1:3] <- c("source","target","pathway_name")' or access all significant interactions using the function `subsetCommunication`

object@netP$pathways list all the signaling pathways with significant communications.

From version >= 1.1.0, pathways are ordered based on the total communication probabilities. NB: pathways with small total communication probabilities might be also very important since they might be specifically activated between only few cell types.

`computeEigengap` *Compute the eigengap of a given matrix for inferring the number of clusters*

Description

Compute the eigengap of a given matrix for inferring the number of clusters

Usage

```
computeEigengap(CM, tau = NULL, tol = 0.01)
```

Arguments

`CM`	consensus matrix
`tau`	truncated consensus matrix
`tol`	tolerance

`computeEnrichmentScore` *Compute and visualize the enrichment score of ligand-receptor pairs in one condition compared to another condition*

Description

Compute and visualize the enrichment score of ligand-receptor pairs in one condition compared to another condition

Usage

```
computeEnrichmentScore(
  df,
  measure = c("ligand", "signaling", "LR-pair"),
  variable.both = TRUE,
  species = c("mouse", "human"),
  db = NULL,
  color.use = NULL,
  color.name = "Dark2",
  n.color = 8,
  scale = c(4, 0.8),
  min.freq = 0,
  max.words = 200,
  random.order = FALSE,
  rot.per = 0,
  return.data = FALSE,
```

```
  seed = 1,
  ...
)
```

Arguments

df	a dataframe
measure	compute the enrichment score in terms of "ligand", "signaling",or "LR-pair"
variable.both	variable.both = TRUE will require that both ligand and receptor from one pair are over-expressed; variable.both = FALSE will only require that either ligand or receptor from one pair is over-expressed.
species	define the species as one of the c('mouse','human') to extract the CellChatDB; For other species, users need to provide a ligand-receptor database 'db'
db	a customized ligand-receptor database 'db'
color.use	defining the color for each group of datasets
color.name	the color names in RColorBrewer::brewer.pal
n.color	the number of colors
scale	A vector of length 2 indicating the range of the size of the words.
min.freq	words with frequency below min.freq will not be plotted
max.words	Maximum number of words to be plotted. least frequent terms dropped
random.order	plot words in random order. If false, they will be plotted in decreasing frequency
rot.per	proportion words with 90 degree rotation
return.data	whether return the data frame for plotting wordcloud
seed	set a seed
...	Other parameters passing to wordcloud::wordcloud

Value

A ggplot object

`computeExpr_agonist` *Modeling the effect of agonist on the ligand-receptor interaction*

Description

Modeling the effect of agonist on the ligand-receptor interaction

Usage

```
computeExpr_agonist(data.use, pairLRsig, cofactor_input, index.agonist, Kh, n)
```

Arguments

data.use	data matrix
pairLRsig	the L-R interactions
cofactor_input	the cofactor_input from CellChatDB
index.agonist	the index of agonist in the database
Kh	a parameter in Hill function
n	Hill coefficient

`computeExpr_antagonist` *Modeling the effect of antagonist on the ligand-receptor interaction*

Description

Modeling the effect of antagonist on the ligand-receptor interaction

Usage

```
computeExpr_antagonist(
  data.use,
  pairLRsig,
  cofactor_input,
  index.antagonist,
  Kh,
  n
)
```

Arguments

data.use	data matrix
pairLRsig	the L-R interactions
cofactor_input	the cofactor_input from CellChatDB
index.antagonist	the index of antagonist in the database
Kh	a parameter in Hill function
n	Hill coefficient

computeExpr_complex *Compute the expression of complex in individual cells using geometric mean*

Description

Compute the expression of complex in individual cells using geometric mean

Usage

```
computeExpr_complex(complex_input, data.use, complex)
```

Arguments

complex_input	the complex_input from CellChatDB
data.use	data matrix (row are genes and columns are cells or cell groups)
complex	the names of complex

computeExpr_coreceptor *Modeling the effect of coreceptor on the ligand-receptor interaction*

Description

Modeling the effect of coreceptor on the ligand-receptor interaction

Usage

```
computeExpr_coreceptor(cofactor_input, data.use, pairLRsig, type = c("A", "I"))
```

Arguments

cofactor_input	the cofactor_input from CellChatDB
data.use	data matrix
pairLRsig	a data frame giving ligand-receptor interactions
type	when type == "A", computing expression of co-activation receptor; when type == "I", computing expression of co-inhibition receptor.

computeExpr_LR *Compute the expression of ligands or receptors using geometric mean*

Description

Compute the expression of ligands or receptors using geometric mean

Usage

```
computeExpr_LR(geneLR, data.use, complex_input)
```

Arguments

geneLR	a char vector giving a set of ligands or receptors
data.use	data matrix (row are genes and columns are cells or cell groups)
complex_input	the complex_input from CellChatDB

computeExprGroup_agonist *Modeling the effect of agonist on the ligand-receptor interaction*

Description

Modeling the effect of agonist on the ligand-receptor interaction

Usage

```
computeExprGroup_agonist(
  data.use,
  pairLRsig,
  cofactor_input,
  group,
  index.agonist,
  Kh,
  FunMean,
  n
)
```

Arguments

data.use	data matrix
pairLRsig	the L-R interactions
cofactor_input	the cofactor_input from CellChatDB
group	a factor defining the cell groups

index.agonist	the index of agonist in the database
Kh	a parameter in Hill function
FunMean	the function for computing mean expression per group
n	Hill coefficient

`computeExprGroup_antagonist` *Modeling the effect of antagonist on the ligand-receptor interaction*

Description

Modeling the effect of antagonist on the ligand-receptor interaction

Usage

```
computeExprGroup_antagonist(
  data.use,
  pairLRsig,
  cofactor_input,
  group,
  index.antagonist,
  Kh,
  FunMean,
  n
)
```

Arguments

data.use	data matrix
pairLRsig	the L-R interactions
cofactor_input	the cofactor_input from CellChatDB
group	a factor defining the cell groups
index.antagonist	the index of antagonist in the database
Kh	a parameter in Hill function
FunMean	the function for computing mean expression per group
n	Hill coefficient

computeLaplacian *Compute eigenvalues of associated Laplacian matrix of a given matrix*

Description

Compute eigenvalues of associated Laplacian matrix of a given matrix

Usage

```
computeLaplacian(CM, tol = 0.01)
```

Arguments

CM	consensus matrix
tol	tolerance

computeNetD_structure *Compute the structural distance between two signaling networks*

Description

Compute the structural distance between two signaling networks

Usage

```
computeNetD_structure(g, h, w1 = 0.45, w2 = 0.45, w3 = 0.1)
```

Arguments

g	a graph object of one signaling network
h	a graph object of another signaling network
w1	parameter
w2	parameter
w3	parameter

computeNetSimilarity *Compute signaling network similarity for any pair of signaling networks*

Description

Compute signaling network similarity for any pair of signaling networks

Usage

```
computeNetSimilarity(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  k = NULL,
  thresh = NULL
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
k	the number of nearest neighbors
thresh	the fraction (0 to 0.25) of interactions to be trimmed before computing network similarity

computeNetSimilarityPairwise *Compute signaling network similarity for any pair of datasets*

Description

Compute signaling network similarity for any pair of datasets

Usage

```
computeNetSimilarityPairwise(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  comparison = NULL,
  k = NULL,
  thresh = NULL
)
```

Arguments

object	A merged CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
comparison	a numerical vector giving the datasets for comparison
k	the number of nearest neighbors
thresh	the fraction (0 to 0.25) of interactions to be trimmed before computing network similarity

---

computeRegionDistance *Compute the region distance based on the spatial locations of each splot/cell of the spatial transcriptomics*

Description

Compute the region distance based on the spatial locations of each splot/cell of the spatial transcriptomics

Usage

```
computeRegionDistance(
  coordinates,
  meta,
  interaction.range = NULL,
  ratio = NULL,
  tol = NULL,
  k.min = 10,
  contact.dependent = TRUE,
  contact.range = NULL,
  contact.knn.k = NULL,
  do.symmetric = TRUE
)
```

Arguments

coordinates	a data matrix in which each row gives the spatial locations/coordinates of each cell/spot
meta	a data frame including at least two columns named ‘group‘ and ‘samples‘. ‘meta$group‘ is a factor vector defining the regions/labels of each cell/spot. ‘meta$samples‘ is a factor vector defining the sample labels of each dataset.
interaction.range	The maximum interaction/diffusion range of ligands. This hard threshold is used to filter out the connections between spatially distant regions

`ratio` a numerical vector giving the conversion factor when converting spatial coordinates from Pixels or other units to Micrometers (i.e.,Microns).

For example, setting 'ratio = 0.18' indicates that 1 pixel equals 0.18um in the coordinates. For 10X visium, it is the ratio of the theoretical spot size (i.e., 65um) over the number of pixels that span the diameter of a theoretical spot size in the full-resolution image (i.e., 'spot.size.fullres' in the 'scalefactors_json.json' file).

`tol` a numerical vector giving the tolerance factor to increase the robustness when comparing the center-to-center distance against the 'interaction.range'. This can be the half value of cell/spot size in the unit of um.

For example, for 10X visium, 'tol' can be set as '65/2'; for slide-seq, 'tol' can be set as '10/2'. If the cell/spot size is not known, we provide a function 'computeCellDistance' to compute the center-to-center distance. 'tol' can be the the half value of the minimum center-to-center distance.

`k.min` the minimum number of interacting cell pairs required for defining adjacent cell groups

`contact.dependent`

Whether determining spatially proximal cell groups based on either the contact.range or the k-nearest neighbors (knn). By default 'contact.dependent = TRUE' when inferring contact-dependent and juxtacrine signaling (including ECM-Receptor and Cell-Cell Contact signaling classified in CellChatDB$interaction$annotation). If only focusing on 'Secreted Signaling', the 'contact.dependent' will be automatically set as FALSE except for 'contact.dependent.forced = TRUE'.

`contact.range` The interaction range (Unit: microns) to restrict the contact-dependent signaling. For spatial transcriptomics in a single-cell resolution, 'contact.range' is approximately equal to the estimated cell diameter (i.e., the cell center-to-center distance), which means that contact-dependent and juxtacrine signaling can only happens when the two cells are contact to each other.

Typically, 'contact.range = 10', which is a typical human cell size. However, for low-resolution spatial data such as 10X visium, it should be the cell center-to-center distance (i.e., 'contact.range = 100' for visium data). The function 'computeCellDistance' can compute the center-to-center distance.

`contact.knn.k` Number of neighbors to restrict the contact-dependent signaling within the neatest neighbors. By default, CellChat uses 'contact.range' to restrict the contact-dependent signaling; however, users can also provide a value of 'contact.knn.k', in order to determine spatially proximal cell groups based on the k-nearest neighbors (knn). For 10X visium, contact.knn.k = 6. For other spatial technologies, this value may be hard to determine because the sequenced cells/spots are usually not regularly arranged.

`do.symmetric` Whether converting the adjacent matrix into symmetric one when determining spatially proximal cell groups. Default is TRUE, indicating that if adj(i,j) or adj(j,i) is zero, then both are zeros.

Value

A list including a square matrix giving the pairwise region distances and an adjacent matrix indicating physically contacting cell groups based on either the contact.range or the k-nearest neighbors

createCellChat *Create a new CellChat object from a data matrix, Seurat or SingleCellExperiment object*

Description

Create a new CellChat object from a data matrix, Seurat or SingleCellExperiment object

Usage

```
createCellChat(
  object,
  meta = NULL,
  group.by = NULL,
  datatype = c("RNA", "spatial"),
  coordinates = NULL,
  spatial.factors = NULL,
  assay = NULL,
  do.sparse = T
)
```

Arguments

object	a normalized (NOT count) data matrix (genes by cells), Seurat or SingleCellExperiment object
meta	a data frame (rows are cells with rownames) consisting of cell information, which will be used for defining cell groups. If input is a Seurat or SingleCellExperiment object, the meta data in the object will be used
group.by	a char name of the variable in meta data, defining cell groups. If input is a data matrix and group.by is NULL, the input 'meta' should contain a column named 'labels', If input is a Seurat or SingleCellExperiment object, USER must provide 'group.by' to define the cell groups. e.g, group.by = "ident" for Seurat object
datatype	By default datatype = "RNA"; when running CellChat on spatial imaging data, set datatype = "spatial" and input 'spatial.factors'
coordinates	a data matrix in which each row gives the spatial locations/coordinates of each cell/spot
spatial.factors	a data frame containing two distance factors 'ratio' and 'tol', which is dependent on spatial transcriptomics technologies (and specific datasets). USER must input this data frame when datatype = "spatial". spatial.factors must contain an element named 'ratio', which is the conversion factor when converting spatial coordinates from Pixels or other units to Micrometers (i.e.,Microns). For example, setting 'ratio = 0.18' indicates that 1 pixel equals 0.18um in the coordinates,

and another element named 'tol', which is the tolerance factor to increase the robustness when comparing the center-to-center distance against the 'interaction.range'. This can be the half value of cell/spot size in the unit of um. If the cell/spot size is not known, we provide a function 'computeCellDistance' to compute the cell center-to-center distance. 'tol' can be the the half value of the minimum center-to-center distance. Of note, CellChat does not need an accurate tolerance factor, which is used for determining whether considering the cell-pair as spatially proximal if their distance is greater than 'interaction.range' but smaller than "'interaction.range' + 'tol'".

`assay` Assay to use when the input is a Seurat or SingleCellExperiment object. NB: The data in the 'integrated' assay in Seurat is not suitable for CellChat analysis because it contains negative values.

`do.sparse` whether use sparse format

Examples

```
## Not run:
Create a CellChat object from single-cell transcriptomics data
# Input is a data matrix
## create a dataframe consisting of the cell labels
meta = data.frame(labels = cell.labels, row.names = names(cell.labels))
cellChat <- createCellChat(object = data.input, meta = meta, group.by = "labels")

# input is a Seurat object
## use the default cell identities of Seurat object
cellChat <- createCellChat(object = seurat.obj, group.by = "ident", assay = "RNA")
## use other meta information as cell groups
cellChat <- createCellChat(object = seurat.obj, group.by = "seurat.clusters")

# input is a SingleCellExperiment object
cellChat <- createCellChat(object = sce.obj, group.by = "sce.clusters")

# input is a AnnData object
sce <- zellkonverter::readH5AD(file = "adata.h5ad")
assayNames(sce) # retrieve all the available assays within sce object
counts <- assay(sce, "X") # add a new assay entry "logcounts" if not available and make sure this is the original cou
library.size <- Matrix::colSums(counts)
logcounts(sce) <- log1p(Matrix::t(Matrix::t(counts)/library.size) * 10000)
meta <- as.data.frame(SingleCellExperiment::colData(sce))
cellChat <- createCellChat(object = sce, group.by = "sce.clusters")


Create a CellChat object from spatial transcriptomics data
# Input is a data matrix
cellChat <- createCellChat(object = data.input, meta = meta, group.by = "labels",
                  datatype = "spatial", coordinates = coordinates, spatial.factors = spatial.factors)

# input is a Seurat object
cellChat <- createCellChat(object = seurat.obj, group.by = "ident", assay = "SCT",
                           datatype = "spatial", spatial.factors = spatial.factors)
```

```
## End(Not run)
```

`dotPlot` *Dot plot*

Description

The size of the dot encodes the percentage of cells within a class, while the color encodes the AverageExpression level across all cells within a class

Usage

```
dotPlot(
  object,
  features,
  rotation = TRUE,
  colormap = "OrRd",
  color.direction = 1,
  color.use = c("#F8766D", "#00BFC4"),
  scale = TRUE,
  col.min = -2.5,
  col.max = 2.5,
  dot.scale = 6,
  assay = "RNA",
  idents = NULL,
  group.by = NULL,
  split.by = NULL,
  legend.width = 0.5,
  angle.x = 45,
  hjust.x = 1,
  angle.y = 0,
  hjust.y = 0.5,
  show.legend = TRUE,
  ...
)
```

Arguments

`object`	seurat object
`features`	Features to plot (gene expression, metrics)
`rotation`	whether rotate the plot
`colormap`	RColorbrewer palette to use (check available palette using RColorBrewer::display.brewer.all()). default will use customed color palette
`color.direction`	Sets the order of colours in the scale. If 1, the default, colours are as output by RColorBrewer::brewer.pal(). If -1, the order of colours is reversed.

color.use	defining the color for each condition/dataset
scale	whther show x-axis text
col.min	Minimum scaled average expression threshold (everything smaller will be set to this)
col.max	Maximum scaled average expression threshold (everything larger will be set to this)
dot.scale	Scale the size of the points, similar to cex
assay	Name of assay to use, defaults to the active assay
idents	Which classes to include in the plot (default is all)
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); pass 'ident' to group by identity class
split.by	Name of a metadata column to split plot by;
legend.width	legend width
angle.x	angle for x-axis text rotation
hjust.x	adjust x axis text
angle.y	angle for y-axis text rotation
hjust.y	adjust y axis text
show.legend	whether show the legend
...	Extra parameters passed to DotPlot from Seurat package

Value

ggplot2 object

`entropia` *compute the Shannon entropy*

Description

compute the Shannon entropy

Usage

```
entropia(a)
```

Arguments

a	a numeric vector

`extract_max` *extract the max value of the y axis*

Description

extract the max value of the y axis

Usage

```
extract_max(p)
```

Arguments

`p`	ggplot object

`extractEnrichedLR` *Identify all the significant interactions (L-R pairs) and related signaling genes for a given signaling pathway*

Description

Identify all the significant interactions (L-R pairs) and related signaling genes for a given signaling pathway

Usage

```
extractEnrichedLR(
  object,
  signaling,
  geneLR.return = FALSE,
  enriched.only = TRUE,
  thresh = 0.05,
  geneInfo = NULL,
  complex_input = NULL
)
```

Arguments

`object`	CellChat object
`signaling`	a char vector containing signaling pathway names for searching
`geneLR.return`	whether return the related signaling genes of enriched L-R pairs
`enriched.only`	whether only return the identified enriched signaling genes in the database. Default = TRUE, returning the significantly enriched signaling interactions
`thresh`	threshold of the p-value for determining significant interaction
`geneInfo`	a dataframe with gene official symbol (there should be one column named 'Symbol')
`complex_input`	signaling complex information from CellChatDB

Value

The returned value depends on the input argument:

When 'geneLR.return = FALSE', it returns a data frame containing the significant interactions (L-R pairs)

When 'geneLR.return = TRUE', it returns a list, the first element is a data frame containing the significant interactions (L-R pairs), and the second is a vector containing the related signaling genes of enriched L-R pairs, which can be used for examining the gene expression pattern using the function `plotGeneExpression`

`extractEnrichedLR_internal` *Identify all the significant interactions (L-R pairs) and related signaling genes for a given signaling pathway*

Description

Identify all the significant interactions (L-R pairs) and related signaling genes for a given signaling pathway

Usage

```
extractEnrichedLR_internal(
  net,
  LR,
  DB,
  signaling,
  enriched.only = TRUE,
  thresh = 0.05
)
```

Arguments

`net, LR, DB`	object@net object@LR object@DB
`signaling`	a char vector containing signaling pathway names for searching
`enriched.only`	whether only return the identified enriched signaling genes in the database. Default = TRUE, returning the significantly enriched signaling interactions
`thresh`	threshold of the p-value for determining significant interaction

Value

a list: list(geneLR, pairLR.name.use)

extractGene *Extract the genes involved in CellChatDB*

Description

Extract the genes involved in CellChatDB

Usage

```
extractGene(CellChatDB)
```

Arguments

CellChatDB	CellChatDB databse used in the analysis

extractGeneSubset *Extract the gene name*

Description

Extract the gene name

Usage

```
extractGeneSubset(geneSet, complex_input, geneIfo)
```

Arguments

geneSet	gene set
complex_input	complex in CellChatDB databse
geneIfo	official gene symbol

extractGeneSubsetFromPair *Extract the signaling gene names from ligand-receptor pairs*

Description

Extract the signaling gene names from ligand-receptor pairs

Usage

```
extractGeneSubsetFromPair(
  pairLR,
  object = NULL,
  complex_input = NULL,
  geneInfo = NULL,
  combined = TRUE
)
```

Arguments

pairLR	data frame must contain columns named 'ligand' and 'receptor'
object	a CellChat object
complex_input	complex in CellChatDB databse
geneInfo	official gene symbol
combined	whether combining the ligand genes and receptor genes

extractLRfromGenes *Extract L-R pairs associated with a given gene set*

Description

Extract L-R pairs associated with a given gene set

Usage

```
extractLRfromGenes(geneSet, db)
```

Arguments

geneSet	a vector of genes
db	one of the CellChatDB databases (e.g., CellChatDB.human, CellChatDB.mouse...)

`filterCommunication` *Filter cell-cell communication if there are only few number of cells in certain cell groups or inconsistent cell-cell communication across samples*

Description

Filter cell-cell communication if there are only few number of cells in certain cell groups or inconsistent cell-cell communication across samples

Usage

```
filterCommunication(
  object,
  min.cells = 10,
  min.samples = NULL,
  rare.keep = FALSE,
  nonFilter.keep = FALSE
)
```

Arguments

`object`	CellChat object
`min.cells`	The minmum number of cells required in each cell group for cell-cell communication
`min.samples`	The minmum number of samples required for consistent cell-cell communication across samples (that is an interaction present in at least 'min.samples' samples) when mutiple samples/replicates/batches are merged as an input for CellChat analysis.
`rare.keep`	Whether to keep the interactions associated with the rare populations when min.samples >= 2. When a rare population is identified in the merged samples (say 15 cells in this rare population from two samples), it is likely to filter out the interactions associated with this rare population when setting min.samples >= 2. Setting 'rare.keep = TRUE' to retain the identified interactions associated with this rare population.
`nonFilter.keep`	Whether to keep the non-filtered cell-cell communication in the CellChat object. This is useful for avoiding re-running 'computeCommunProb' if you want to adjust the parameters when running 'filterCommunication'.

Value

CellChat object with an updated slot net

findEnrichedSignaling *Find the enriched signaling according to the genes (e.g.DEGs) and cell groups of interest*

Description

Find the enriched signaling according to the genes (e.g.DEGs) and cell groups of interest

Usage

```
findEnrichedSignaling(
  object,
  features,
  idents = NULL,
  pattern = c("both", "outgoing", "incoming"),
  thresh = 0.05
)
```

Arguments

object	CellChat object
features	a vector giving the genes of interest
idents	a vector giving the names of cell groups of interest. If idents = NULL, it returns signaling according to the input features.
pattern	"both", "outgoing" or "incoming"
thresh	threshold of the p-value for determining significant interaction

Value

a dataframe of the cell-cell communication associated with the input features.

Examples

```
## Not run:
# find all the significant outgoing signaling according to the features and cell groups of interest
df <- findEnrichedSignaling(object, features = c("CCL19", "CXCL12"), idents = c("Inflam. FIB", "COL11A1+ FIB"), pat

## End(Not run)
```

geometricMean *Compute the geometric mean*

Description

Compute the geometric mean

Usage

```
geometricMean(x, na.rm = TRUE)
```

Arguments

x	a numeric vector
na.rm	whether remove na

getMaxWeight *Compute the maximum value of certain measures in the inferred cell-cell communication networks*

Description

To better control the node size and edge weights of the inferred networks across different datasets, we compute the maximum number of cells per cell group and the maximum number of interactions (or interaction weights) across all datasets

Usage

```
getMaxWeight(
  object.list,
  slot.name = c("idents", "net"),
  attribute = c("idents", "count")
)
```

Arguments

object.list	List of CellChat objects
slot.name	the slot name of object that is used to compute the maximum value. When slot.name = "idents", 'attribute' should be "idents", which will compute the maximum number of cells per cell group across all datasets When slot.name = "net", 'attribute' can be either "count" or "weight", which will compute he maximum number of interactions (or interaction weights) across all datasets When slot.name = "net" or "netP", 'attribute' can be a single pathway name or a ligand-receptor pair name

attribute	the attribute to compute the maximum values. ‘attribute‘ should have the same length as ‘slot.name‘. ‘attribute‘ can only be "count", "weight","count.merged","weight.merged" or a single pathway name or a ligand-receptor pair name

Value

A numeric vector

`ggPalette` *Generate ggplot2 colors*

Description

Generate ggplot2 colors

Usage

```
ggPalette(n)
```

Arguments

n	number of colors to generate

`identifyCommunicationPatterns` *Identification of major signals for specific cell groups and general communication patterns*

Description

Identification of major signals for specific cell groups and general communication patterns

Usage

```
identifyCommunicationPatterns(
  object,
  slot.name = "netP",
  pattern = c("outgoing", "incoming"),
  k = NULL,
  k.range = seq(2, 10),
  heatmap.show = TRUE,
  color.use = NULL,
  color.heatmap = "Spectral",
  title.legend = "Contributions",
  width = 4,
```

```
  height = 6,
  font.size = 8
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
pattern	"outgoing" or "incoming"
k	the number of patterns
k.range	a range of the number of patterns
heatmap.show	whether showing heatmap
color.use	the character vector defining the color of each cell group
color.heatmap	a color name in brewer.pal
title.legend	the title of legend in heatmap
width	width of heatmap
height	height of heatmap
font.size	fontsize in heatmap

---

identifyEnrichedInteractions

Identify all the significant interactions (L-R pairs) from some cell groups to other cell groups

---

Description

Identify all the significant interactions (L-R pairs) from some cell groups to other cell groups

Usage

```
identifyEnrichedInteractions(
  object,
  from,
  to,
  bidirection = FALSE,
  pair.only = TRUE,
  pairLR.use0 = NULL,
  thresh = 0.05
)
```

Arguments

object	CellChat object
from	a vector giving the index or the name of source cell groups
to	a corresponding vector giving the index or the name of target cell groups. Note: The length of 'from' and 'to' must be the same, giving the corresponding pair of cell groups for communication.
bidirection	whether show the bidirectional communication, i.e., both 'from'->'to' and 'to'->'from'.
pair.only	whether only return ligand-receptor pairs without pathway names and communication strength
pairLR.use0	ligand-receptor pairs to use; default is all the significant interactions
thresh	threshold of the p-value for determining significant interaction

---

identifyOverExpressedGenes *Identify over-expressed signaling genes associated with each cell group*

Description

USERS can use customized gene set as over-expressed signaling genes by setting 'object@var.features[[features.name]] <- features.sig' The Bonferroni corrected/adjusted p value can be obtained via 'object@var.features[[paste0(features.name, ".info")]]'. Note that by default 'features.name = "features"'

Usage

```
identifyOverExpressedGenes(
  object,
  data.use = NULL,
  group.by = NULL,
  idents.use = NULL,
  invert = FALSE,
  group.dataset = NULL,
  pos.dataset = NULL,
  group.DE.combined = FALSE,
  features.name = "features",
  only.pos = TRUE,
  features = NULL,
  return.object = TRUE,
  thresh.pc = 0,
  thresh.fc = 0,
  thresh.p = 0.05,
  do.DE = TRUE,
  do.fast = TRUE,
  min.cells = 10
)
```

Arguments

object	CellChat object
data.use	a customed data matrix. Default: data.use = NULL and the expression matrix in the slot 'data.signaling' is used
group.by	cell group information; default is 'object@idents'; otherwise it should be one of the column names of the meta slot
idents.use	a subset of cell groups used for analysis
invert	whether to invert the idents.use
group.dataset	dataset origin information in a merged CellChat object; set it as one of the column names of meta slot when identifying the highly enriched genes in one dataset for each cell group
pos.dataset	the dataset name used for identifying highly enriched genes in this dataset for each cell group
group.DE.combined	Whether to perform differential expression between conditions by ignoring cell group information. By default, group.DE.combined = FALSE, which will perform differential expression analysis between two biological conditions for each cell group; When group.DE.combined = TRUE, it will perform DE analysis by combining all cell groups together.
features.name	a char name used for storing the over-expressed signaling genes in 'object@var.features[[features.name]]'
only.pos	Only return positive markers
features	features used for identifying Over Expressed genes. default use all features
return.object	whether to return the object; otherwise return a data frame consisting of over-expressed signaling genes associated with each cell group
thresh.pc	Threshold of the fraction of cells expressed in one cluster, i.e., thresh.pc = 0.1
thresh.fc	Threshold of Log Fold Change, i.e., thresh.pc = 0.1
thresh.p	Threshold of p-values, i.e., thresh.pc = 0.05
do.DE	Whether to perform differential expression analysis. By default do.DE = TRUE; When do.DE = FALSE, selecting over-expressed genes that are expressed in more than 'min.cells' cells.
do.fast	If do.fast = TRUE, then perform a ultra-fast Wilcoxon test using presto package; otherwise using stats package. These two methods produce different logFC values, and the presto::wilcoxauc method gives smaller values.
min.cells	the minmum number of expressed cells required for the genes that are considered for cell-cell communication analysis

Value

A CellChat object or a data frame. If returning a CellChat object, two new elements named 'features.name' and paste0(features.name, ".info") will be added into the list 'object@var.features' 'object@var.features[[features.name]]' is a vector consisting of the identified over-expressed signaling genes; 'object@var.features[[paste0(features.name, ".info")]]' is a data frame returned from the differential expression analysis

`identifyOverExpressedInteractions` *Identify over-expressed ligand-receptor interactions (pairs) within the used CellChatDB*

Description

Identify over-expressed ligand-receptor interactions (pairs) within the used CellChatDB

Usage

```
identifyOverExpressedInteractions(
  object,
  features.name = "features",
  variable.both = TRUE,
  features = NULL,
  return.object = TRUE
)
```

Arguments

`object`	CellChat object
`features.name`	a char name used for assess the results in 'object@var.features[[features.name]]'
`variable.both`	variable.both = TRUE will require that both ligand and receptor from one pair are over-expressed; variable.both = FALSE will only require that either ligand or receptor from one pair is over-expressed, leading to more over-expressed ligand-receptor interactions (pairs) for further analysis.
`features`	a vector of features to use. default use all over-expressed genes in 'object@var.features[[features.name]]'
`return.object`	whether returning a CellChat object. If FALSE, it will return a data frame containing the over-expressed ligand-receptor pairs

Value

A CellChat object or a data frame. If returning a CellChat object, a new element named 'LRsig' will be added into the list 'object@LR'

identifyOverExpressedLigandReceptor

Identify over-expressed ligands and (complex) receptors associated with each cell group

Description

This function identifies the over-expressed ligands and (complex) receptors based on the identified signaling genes from 'identifyOverExpressedGenes'.

Usage

```
identifyOverExpressedLigandReceptor(
  object,
  features.name = "features",
  features = NULL,
  return.object = TRUE
)
```

Arguments

object	CellChat object
features.name	a char name used for storing the over-expressed ligands and receptors in 'object@var.features[[paste0(features.name, ".LR")]]'
features	a vector of features to use. default use all over-expressed genes in 'object@var.features[[features.name]]'
return.object	whether returning a CellChat object. If FALSE, it will return a data frame containing over-expressed ligands and (complex) receptors associated with each cell group

Value

A CellChat object or a data frame. If returning a CellChat object, a new element named paste0(features.name, ".LR") will be added into the list 'object@var.features'

liftCellChat

Update a CellChat object by lifting up the cell groups to the same cell labels across all datasets

Description

This function is useful when comparing inferred communications across different datasets with different cellular compositions

Usage

```
liftCellChat(object, group.new = NULL)
```

Arguments

object	A single or merged CellChat object
group.new	A char vector giving the cell labels to lift up. The order of cell labels in the vector will be used for setting the new cell identity. If the input is a merged CellChat object and group.new = NULL, it will use the cell labels from one dataset with the maximum number of cell groups If the input is a single CellChat object, 'group.new' must be defined.

Value

a updated CellChat object

---

mergeCellChat *Merge CellChat objects*

Description

Merge CellChat objects

Usage

```
mergeCellChat(
  object.list,
  add.names = NULL,
  merge.data = FALSE,
  cell.prefix = FALSE
)
```

Arguments

object.list	A list of multiple CellChat objects
add.names	A vector containing the name of each dataset
merge.data	whether merging the data for ALL genes. Default only merges the data of signaling genes
cell.prefix	whether prefix cell names

`mergeInteractions` *Compute the number of interactions/interaction strength between cell types based on their associated cell subpopulations*

Description

Compute the number of interactions/interaction strength between cell types based on their associated cell subpopulations

Usage

```
mergeInteractions(object, group.merged)
```

Arguments

`object`	CellChat object
`group.merged`	a factor defining the group for merging different clusters/subpopulations

Value

An updated slot 'net' by adding three elements:

'count.merged': the number of interactions between cell types (i.e., merged cell groups)

'weight.merged': interaction strength between cell types (i.e., merged cell groups)

'group.merged' the defined group for merging different clusters/subpopulations

`modify_vlnplot` *modified vlnplot*

Description

modified vlnplot

Usage

```
modify_vlnplot(
  object,
  features,
  idents = NULL,
  split.by = NULL,
  cols = NULL,
  show.text.y = TRUE,
  line.size = NULL,
  pt.size = 0,
  plot.margin = margin(0, 0, 0, 0, "cm"),
  ...
)
```

Arguments

object	Seurat object
features	Features to plot (gene expression, metrics)
idents	Which classes to include in the plot (default is all)
split.by	Name of a metadata column to split plot by;
cols	defining the color for each cell group
show.text.y	whther show y-axis text
line.size	line width in the violin plot
pt.size	size of the dots
plot.margin	adjust the white space between each plot
...	pass any arguments to VlnPlot in Seurat

---

mycircle *generate circle symbol*

Description

generate circle symbol

Usage

```
mycircle(coords, v = NULL, params)
```

Arguments

coords	coordinates of points
v	vetex
params	parameters

---

netAnalysis_computeCentrality *Compute the network centrality scores allowing identification of dominant senders, receivers, mediators and influencers in all inferred communication networks*

Description

NB: This function was previously named as ‘netAnalysis_signalingRole‘. The previous function ‘netVisual_signalingRole‘ is now named as ‘netAnalysis_signalingRole_network‘.

Usage

```
netAnalysis_computeCentrality(
  object = NULL,
  slot.name = "netP",
  net = NULL,
  net.name = NULL,
  thresh = 0.05
)
```

Arguments

object	CellChat object; If object = NULL, USER must provide 'net'
slot.name	the slot name of object that is used to compute centrality measures of signaling networks. Setting slot.name = "netP" to compute the network centrality scores at the level of signaling pathways, and setting slot.name = "net" to compute the network centrality scores at the level of ligand-receptor pairs
net	compute the centrality measures on a specific signaling network given by a 2 or 3 dimemsional array net
net.name	a character vector giving the name of signaling networks
thresh	threshold of the p-value for determining significant interaction

---

netAnalysis_contribution

Compute and visualize the contribution of each ligand-receptor pair in the overall signaling pathways

---

Description

Compute and visualize the contribution of each ligand-receptor pair in the overall signaling pathways

Usage

```
netAnalysis_contribution(
  object,
  signaling,
  signaling.name = NULL,
  sources.use = NULL,
  targets.use = NULL,
  width = 0.1,
  vertex.receiver = NULL,
  thresh = 0.05,
  return.data = FALSE,
  x.rotation = 0,
  title = "Contribution of each L-R pair",
```

```
  font.size = 10,
  font.size.title = 10
)
```

Arguments

object	CellChat object
signaling	a signaling pathway name
signaling.name	alternative signaling pathway name to show on the plot
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
width	the width of individual bar
vertex.receiver	a numeric vector giving the index of the cell groups as targets in the first hierarchy plot
thresh	threshold of the p-value for determining significant interaction
return.data	whether return the data.frame consisting of the predicted L-R pairs and their contribution
x.rotation	rotation of x-label
title	the title of the plot
font.size	font size of the text
font.size.title	font size of the title

---

netAnalysis_diff_signalingRole_scatter

2D visualization of differential signaling roles (dominant senders (sources) or receivers (targets)) of each cell group when comparing mutiple datasets

---

Description

This scatter plot shows the differential signaling roles (dominant senders (sources) or receivers (targets) in a 2D space.

x-axis and y-axis are respectively the differential outgoing or incoming communication probability associated with each cell group. Dot colors indicate different cell groups. Dot shapes indicate different categories of cell groups if ‘group“ is defined.

Positive values indicate the increase in the second dataset while negative values indicate the increase in the first dataset

Usage

```
netAnalysis_diff_signalingRole_scatter(
  object,
  color.use = NULL,
  comparison = c(1, 2),
  signaling = NULL,
  signaling.exclude = NULL,
  idents.exclude = NULL,
  slot.name = "netP",
  group = NULL,
  dot.size = 2.5,
  point.shape = c(21, 22, 24, 23, 25, 8, 3),
  label.size = 3,
  dot.alpha = 0.6,
  x.measure = "outdeg",
  y.measure = "indeg",
  xlabel = "Outgoing interaction strength",
  ylabel = "Incoming interaction strength",
  title = NULL,
  font.size = 10,
  font.size.title = 10,
  do.label = T,
  show.legend = T,
  show.axes = T
)
```

Arguments

object	A merged CellChat object of a list of CellChat objects
color.use	defining the color for each cell group
comparison	an index vector giving the two datasets for comparison
signaling	a char vector containing signaling pathway names. signaling = NULL: Signaling role analysis on the aggregated cell-cell communication network from all signaling pathways
signaling.exclude	signaling pathways to exclude
idents.exclude	cell groups to exclude. This is useful when zooming into the small changes
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
group	a vector to categorize the cell groups, e.g., categorize the cell groups into two major categories: immune cells and fibroblasts
dot.size	the size of the symbol
point.shape	point shape when group is not NULL
label.size	font size of the text
dot.alpha	transparency

x.measure	The measure used as x-axis. This measure should be one of 'names(slot(object, slot.name)$centr[[1]])' computed from 'netAnalysis_computeCentrality' Default = "outdeg" is the weighted outgoing links (i.e., Outgoing interaction strength). If setting as "outdeg_unweighted", it represents the total number of outgoing signaling.
y.measure	The measure used as y-axis. This measure should be one of 'names(slot(object, slot.name)$centr[[1]])' computed from 'netAnalysis_computeCentrality' Default = "indeg" is the weighted incoming links (i.e., Incoming interaction strength). If setting as "indeg_unweighted", it represents the total number of incoming signaling.
xlabel	label of x-axis
ylabel	label of y-axis
title	main title of the plot
font.size	font size of the text
font.size.title	font size of the title
do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

Value

ggplot object

`netAnalysis_dot` *Dot plots showing the associations of latent patterns with cell groups and ligand-receptor pairs or signaling pathways*

Description

Using a contribution score of each cell group to each signaling pathway computed by multiplying W by H obtained from 'identifyCommunicationPatterns', we constructed a dot plot in which the dot size is proportion to the contribution score to show association between cell group and their enriched signaling pathways.

Usage

```
netAnalysis_dot(
  object,
  slot.name = "netP",
  pattern = c("outgoing", "incoming"),
  cutoff = NULL,
  color.use = NULL,
  pathway.show = NULL,
```

```
  group.show = NULL,
  shape = 21,
  dot.size = c(1, 3),
  dot.alpha = 1,
  main.title = NULL,
  font.size = 10,
  font.size.title = 12
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
pattern	"outgoing" or "incoming"
cutoff	the threshold for filtering out weak links. Default is 1/R where R is the number of latent patterns. We set the elements in W and H to be zero if they are less than 'cutoff'.
color.use	the character vector defining the color of each cell group
pathway.show	the character vector defining the signaling to show
group.show	the character vector defining the cell group to show
shape	the shape of the symbol: 21 for circle and 22 for square
dot.size	a range defining the size of the symbol
dot.alpha	transparency
main.title	the title of plot
font.size	font size of the text
font.size.title	font size of the title

---

netAnalysis_river *River plot showing the associations of latent patterns with cell groups and ligand-receptor pairs or signaling pathways*

Description

River (alluvial) plot shows the correspondence between the inferred latent patterns and cell groups as well as ligand-receptor pairs or signaling pathways.

Usage

```
netAnalysis_river(
  object,
  slot.name = "netP",
  pattern = c("outgoing", "incoming"),
  cutoff = 0.5,
  sources.use = NULL,
  targets.use = NULL,
  signaling = NULL,
  color.use = NULL,
  color.use.pattern = NULL,
  color.use.signaling = "grey50",
  do.order = FALSE,
  main.title = NULL,
  font.size = 2.5,
  font.size.title = 12
)
```

Arguments

object	CellChat object
slot.name	the slot name of object: "netP" or "net". Use "netP" to analyze cell-cell communication at the level of signaling pathways, and "net" to analyze cell-cell communication at the level of ligand-receptor pairs.
pattern	"outgoing" or "incoming"
cutoff	the threshold for filtering out weak links
sources.use	a vector giving the index or the name of source cell groups of interest
targets.use	a vector giving the index or the name of target cell groups of interest
signaling	a character vector giving the name of signaling pathways of interest
color.use	the character vector defining the color of each cell group
color.use.pattern	the character vector defining the color of each pattern
color.use.signaling	the character vector defining the color of each signaling
do.order	whether reorder the cell groups or signaling according to their similarity
main.title	the title of plot
font.size	font size of the text
font.size.title	font size of the title

Details

The thickness of the flow indicates the contribution of the cell group or signaling pathway to each latent pattern. The height of each pattern is proportional to the number of its associated cell groups or signaling pathways.

Outgoing patterns reveal how the sender cells coordinate with each other as well as how they coordinate with certain signaling pathways to drive communication.

Incoming patterns show how the target cells coordinate with each other as well as how they coordinate with certain signaling pathways to respond to incoming signaling.

`netAnalysis_signalingChanges_scatter` *2D visualization of differential outgoing and incoming signaling associated with one cell group*

Description

Positive values indicate the increase in the second dataset while negative values indicate the increase in the first dataset

Usage

```
netAnalysis_signalingChanges_scatter(
  object,
  idents.use,
  color.use = c("grey10", "#F8766D", "#00BFC4"),
  comparison = c(1, 2),
  signaling = NULL,
  signaling.label = NULL,
  top.label = 1,
  signaling.exclude = NULL,
  xlims = NULL,
  ylims = NULL,
  slot.name = "netP",
  dot.size = 2.5,
  point.shape = c(21, 22, 24, 23),
  label.size = 3,
  dot.alpha = 0.6,
  x.measure = "outdeg",
  y.measure = "indeg",
  xlabel = "Differential outgoing interaction strength",
  ylabel = "Differential incoming interaction strength",
  title = NULL,
  font.size = 10,
  font.size.title = 10,
  do.label = T,
  show.legend = T,
  show.axes = T
)
```

Arguments

object	A merged CellChat object of a list of CellChat objects
idents.use	the cell group names of interest. Should be one of 'levels(object@idents$joint)'
color.use	a vector with three elements: the first is for coloring shared pathways, the second is for specific pathways in the first dataset, and the third is for specific pathways in the second dataset
comparison	an index vector giving the two datasets for comparison
signaling	a char vector containing signaling pathway names. signaling = NULL: Signaling role analysis on the aggregated cell-cell communication network from all signaling pathways
signaling.label	a char vector giving the signaling names to show when labeling each point
top.label	the fraction of signaling pathways to label
signaling.exclude	signaling pathways to exclude when plotting
xlims, ylims	set x-Axis and y-Axis Limits for zoom into the plot. e.g., xlims = c(-0.05, 0.1), ylims = c(-0.01, 0.035)
slot.name	the slot name of object
dot.size	the size of the symbol
point.shape	point shape
label.size	font size of the text
dot.alpha	transparency
x.measure	The measure used as x-axis. This measure should be one of 'names(slot(object, slot.name)$centr[[1]])' computed from 'netAnalysis_computeCentrality' Default = "outdeg" is the weighted outgoing links (i.e., Outgoing interaction strength). If setting as "outdeg_unweighted", it represents the total number of outgoing signaling.
y.measure	The measure used as y-axis. This measure should be one of 'names(slot(object, slot.name)$centr[[1]])' computed from 'netAnalysis_computeCentrality' Default = "indeg" is the weighted incoming links (i.e., Incoming interaction strength). If setting as "indeg_unweighted", it represents the total number of incoming signaling.
xlabel	label of x-axis
ylabel	label of y-axis
title	main title of the plot
font.size	font size of the text
font.size.title	font size of the title
do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

Value

ggplot object

`netAnalysis_signalingRole_heatmap` *Heatmap showing the contribution of signals (signaling pathways or ligand-receptor pairs) to cell groups in terms of outgoing or incoming signaling*

Description

In this heatmap, colobar represents the relative signaling strength of a signaling pathway across cell groups (NB: values are row-scaled). The top colored bar plot shows the total signaling strength of a cell group by summarizing all signaling pathways displayed in the heatmap. The right grey bar plot shows the total signaling strength of a signaling pathway by summarizing all cell groups displayed in the heatmap.

Usage

```
netAnalysis_signalingRole_heatmap(
  object,
  signaling = NULL,
  pattern = c("outgoing", "incoming", "all"),
  slot.name = "netP",
  color.use = NULL,
  color.heatmap = "BuGn",
  title = NULL,
  width = 10,
  height = 8,
  ylim.top = NULL,
  ylim.right = NULL,
  font.size = 8,
  font.size.title = 10,
  cluster.rows = FALSE,
  cluster.cols = FALSE
)
```

Arguments

`object`	CellChat object
`signaling`	a character vector giving the names of signaling networks of interest
`pattern`	this parameter can be set as "outgoing", "incoming" or "all". When pattern = "all", CellChat aggregates the outgoing and incoming signaling strength together;

slot.name	the slot name of object that is used to examine the signaling patterns at the level of signaling pathways (slot.name = "netP") or ligand-receptor pairs (slot.name = "net");
color.use	the character vector defining the color of each cell group
color.heatmap	a color name in brewer.pal
title	title name
width	width of heatmap
height	height of heatmap
ylim.top	set the range of the top barplot (e.g., ylim.top = c(0, 4))
ylim.right	set the range of the right barplot (e.g., ylim.right = c(0, 5))
font.size	fontsize in heatmap
font.size.title	font size of the title
cluster.rows	whether cluster rows
cluster.cols	whether cluster columns

`netAnalysis_signalingRole_network` *Heatmap showing the centrality scores/importance of cell groups as senders, receivers, mediators and influencers in a single intercellular communication network*

Description

Heatmap showing the centrality scores/importance of cell groups as senders, receivers, mediators and influencers in a single intercellular communication network

Usage

```
netAnalysis_signalingRole_network(
  object,
  signaling,
  slot.name = "netP",
  measure = c("outdeg", "indeg", "flowbet", "info"),
  measure.name = c("Sender", "Receiver", "Mediator", "Influencer"),
  color.use = NULL,
  color.heatmap = "BuGn",
  width = 6.5,
  height = 1.4,
  font.size = 8,
  font.size.title = 10,
  cluster.rows = FALSE,
  cluster.cols = FALSE
)
```

Arguments

object	CellChat object
signaling	a character vector giving the name of signaling networks
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
measure	centrality measures to show
measure.name	the names of centrality measures to show
color.use	the character vector defining the color of each cell group
color.heatmap	a color name in brewer.pal
width	width of heatmap
height	height of heatmap
font.size	fontsize in heatmap
font.size.title	font size of the title
cluster.rows	whether cluster rows
cluster.cols	whether cluster columns

---

netAnalysis_signalingRole_scatter

2D visualization of dominant senders (sources) and receivers (targets)

Description

This scatter plot shows the dominant senders (sources) and receivers (targets) in a 2D space. x-axis and y-axis are respectively the total outgoing or incoming communication probability associated with each cell group. Dot size is proportional to the number of inferred links (both outgoing and incoming) associated with each cell group. Dot colors indicate different cell groups. Dot shapes indicate different categories of cell groups if ‘group“ is defined.

Usage

```
netAnalysis_signalingRole_scatter(
  object,
  signaling = NULL,
  color.use = NULL,
  slot.name = "netP",
  group = NULL,
  weight.MinMax = NULL,
  dot.size = c(2, 6),
  point.shape = c(21, 22, 24, 23, 25, 8, 3),
  label.size = 3,
  dot.alpha = 0.6,
  x.measure = "outdeg",
```

```
  y.measure = "indeg",
  xlabel = "Outgoing interaction strength",
  ylabel = "Incoming interaction strength",
  title = NULL,
  font.size = 10,
  font.size.title = 10,
  do.label = T,
  show.legend = T,
  show.axes = T
)
```

Arguments

object	CellChat object
signaling	a char vector containing signaling pathway names. signaling = NULL: signaling role analysis on the aggregated cell-cell communication network from all signaling pathways
color.use	defining the color for each cell group
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
group	a vector to categorize the cell groups, e.g., categorize the cell groups into two major categories: immune cells and fibroblasts
weight.MinMax	the Minmum/maximum weight, which is useful to control the dot size when comparing multiple datasets
dot.size	a range defining the size of the symbol
point.shape	point shape when group is not NULL
label.size	font size of the text
dot.alpha	transparency
x.measure	The measure used as x-axis. This measure should be one of ‘names(slot(object, slot.name)$centr[[1]])‘ computed from ‘netAnalysis_computeCentrality‘ Default = "outdeg" is the weighted outgoing links (i.e., outgoing interaction strength). If setting as "outdeg_unweighted", it represents the total number of outgoing signaling.
y.measure	The measure used as y-axis. This measure should be one of ‘names(slot(object, slot.name)$centr[[1]])‘ computed from ‘netAnalysis_computeCentrality‘ Default = "indeg" is the weighted incoming links (i.e., incoming interaction strength). If setting as "indeg_unweighted", it represents the total number of incoming signaling.
xlabel	label of x-axis
ylabel	label of y-axis
title	main title of the plot
font.size	font size of the text
font.size.title	font size of the title

do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

Value

ggplot object

`netClustering` *Classification learning of the signaling networks*

Description

Classification learning of the signaling networks

Usage

```
netClustering(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  comparison = NULL,
  k = NULL,
  methods = "kmeans",
  do.plot = TRUE,
  fig.id = NULL,
  do.parallel = TRUE,
  nCores = 4,
  k.eigen = NULL
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
comparison	a numerical vector giving the datasets for comparison. No need to define for a single dataset. Default are all datasets when object is a merged object
k	the number of signaling groups when running kmeans
methods	the methods for clustering: "kmeans" or "spectral"
do.plot	whether showing the eigenspectrum for inferring number of clusters; Default will save the plot
fig.id	add a unique figure id when saving the plot

do.parallel	whether doing parallel when inferring the number of signaling groups when running kmeans
nCores	number of workers when doing parallel
k.eigen	the number of eigenvalues used when doing spectral clustering

---

`netEmbedding` *Manifold learning of the signaling networks based on their similarity*

Description

Manifold learning of the signaling networks based on their similarity

Usage

```
netEmbedding(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  comparison = NULL,
  pathway.remove = NULL,
  umap.method = c("umap-learn", "uwot"),
  n_neighbors = NULL,
  min_dist = 0.3,
  ...
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
comparison	a numerical vector giving the datasets for comparison. No need to define for a single dataset. Default are all datasets when object is a merged object
pathway.remove	a range of the number of patterns
umap.method	UMAP implementation to run. Can be umap-learn: Run the python umap-learn package; uwot: Runs umap via the uwot R package; If umap.method = "uwot", please make sure you have installed the 'uwot' (https://github.com/jlmelville/uwot)
n_neighbors	the number of nearest neighbors in running umap
min_dist	This controls how tightly the embedding is allowed compress points together. Larger values ensure embedded points are moreevenly distributed, while smaller values allow the algorithm to optimise more accurately with regard to local structure. Sensible values are in the range 0.001 to 0.5.
...	Parameters passing to umap

`netMappingDEG` *Mapping the differential expressed genes (DEG) information onto the inferred cell-cell communications*

Description

This function returns a data frame consisting of all the inferred cell-cell communications with mapped DEG information

Usage

```
netMappingDEG(object, features.name, variable.all = TRUE, thresh = 0.05)
```

Arguments

`object`	CellChat object
`features.name`	a char name used for extracting the DEG in 'object@var.features[[features.name]]'
`variable.all`	variable.all = TRUE will compute the c("pvalues", "logFC", "pct.1", "pct.2") for a ligand/receptor complex using the mean value of its all subunits, that is requiring all subunits of the complex are differential expressed; variable.all = FALSE will compute the minimum value of "pvalues" and maximum value of c("logFC", "pct.1", "pct.2") among the subunits, that is only requiring that any one of the subunits of the complex is differential expressed.
`thresh`	threshold of the p-value for determining significant interaction

Value

a data frame of the inferred cell-cell communications, consisting of source, target, interaction_name, pathway_name, prob and other CellChatDB information as well as DEG information

`netVisual` *Visualize the inferred cell-cell communication network*

Description

Automatically save plots in the current working directory.

Usage

```
netVisual(
  object,
  signaling,
  signaling.name = NULL,
  color.use = NULL,
  vertex.receiver = NULL,
```

```
  sources.use = NULL,
  targets.use = NULL,
  top = 1,
  remove.isolate = FALSE,
  vertex.weight = 1,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  weight.scale = TRUE,
  edge.weight.max.individual = NULL,
  edge.weight.max.aggregate = NULL,
  edge.width.max = 8,
  layout = c("circle", "hierarchy", "chord", "spatial"),
  height = 5,
  thresh = 0.05,
  pt.title = 12,
  title.space = 6,
  vertex.label.cex = 0.8,
  from = NULL,
  to = NULL,
  bidirection = NULL,
  vertex.size = NULL,
  out.format = c("svg", "png"),
  sample.use = NULL,
  alpha.image = 0.15,
  point.size = 1.5,
  group = NULL,
  cell.order = NULL,
  small.gap = 1,
  big.gap = 10,
  scale = FALSE,
  reduce = -1,
  show.legend = FALSE,
  legend.pos.x = 20,
  legend.pos.y = 20,
  nCol = NULL,
  ...
)
```

Arguments

object	CellChat object
signaling	a signaling pathway name
signaling.name	alternative signaling pathway name to show on the plot
color.use	the character vector defining the color of each cell group
vertex.receiver	a numeric vector giving the index of the cell groups as targets in the first hierarchy plot
sources.use	a vector giving the index or the name of source cell groups

targets.use	a vector giving the index or the name of target cell groups.
top	the fraction of interactions to show (0 < top <= 1)
remove.isolate	whether remove the isolate nodes in the communication network
vertex.weight	The weight of vertex: either a scale value or a vector
	Default is a scale value being 1, indicating all vertex is plotted in the same size;
	Set 'vertex.weight' as a vector to plot vertex in different size; setting 'vertex.weight = NULL' will have vertex with different size that are portional to the number of cells in each cell group.
vertex.weight.max	
	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	
	the maximum vertex size for visualization
weight.scale	whether scale the edge weight
edge.weight.max.individual	
	the maximum weight of edge when plotting the individual L-R netwrok; defualt = max(net)
edge.weight.max.aggregate	
	the maximum weight of edge when plotting the aggregated signaling pathway network
edge.width.max	The maximum edge width for visualization
layout	"hierarchy", "circle" or "chord"
height	height of plot
thresh	threshold of the p-value for determining significant interaction
pt.title	font size of the text
title.space	the space between the title and plot
vertex.label.cex	
	The label size of vertex in the network
from, to, bidirection	
	Deprecated. Use 'sources.use','targets.use'
vertex.size	Deprecated. Use 'vertex.weight'
	Parameters below are set for "chord" diagram. Please also check the function 'netVisual_chord_cell' for more parameters.
out.format	the format of output figures: svg, png and pdf
	Parameters below are set for "spatial" diagram. Please also check the function 'netVisual_spatial' for more parameters.
sample.use	the sample used for visualization, which should be the element in 'object@meta$samples'.
alpha.image	the transparency of individual spots
point.size	the size of spots
group	A named group labels for making multiple-group Chord diagrams. The sector names should be used as the names in the vector. The order of group controls the sector orders and if group is set as a factor, the order of levels controls the order of groups.

cell.order	a char vector defining the cell type orders (sector orders)
small.gap	Small gap between sectors.
big.gap	Gap between the different sets of sectors, which are defined in the 'group' parameter
scale	scale each sector to same width; default = FALSE; however, it is set to be TRUE when remove.isolate = TRUE
reduce	if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.
show.legend	whether show the figure legend
legend.pos.x, legend.pos.y	adjust the legend position
nCol	number of columns when displaying the network mediated by ligand-receptor using "circle" or "chord"
...	other parameters (e.g.,vertex.label.cex, vertex.label.color, alpha.edge, label.edge, edge.label.color, edge.label.cex, edge.curved, text.x, text.y) passing to 'netVisual_hierarchy1','netVisual_hierarchy2','netVisual_circle'. NB: some parameters might be not supported

---

`netVisual_aggregate` *Visualize the inferred signaling network of signaling pathways by aggregating all L-R pairs*

Description

Visualize the inferred signaling network of signaling pathways by aggregating all L-R pairs

Usage

```
netVisual_aggregate(
  object,
  signaling,
  signaling.name = NULL,
  color.use = NULL,
  thresh = 0.05,
  vertex.receiver = NULL,
  sources.use = NULL,
  targets.use = NULL,
  idents.use = NULL,
  top = 1,
  remove.isolate = FALSE,
  vertex.weight = 1,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  weight.scale = TRUE,
```

```
  edge.weight.max = NULL,
  edge.width.max = 8,
  layout = c("circle", "hierarchy", "chord", "spatial"),
  pt.title = 12,
  title.space = 6,
  vertex.label.cex = 0.8,
  sample.use = NULL,
  alpha.image = 0.15,
  point.size = 1.5,
  group = NULL,
  cell.order = NULL,
  small.gap = 1,
  big.gap = 10,
  scale = FALSE,
  reduce = -1,
  show.legend = FALSE,
  legend.pos.x = 20,
  legend.pos.y = 20,
  ...
)
```

Arguments

object	CellChat object
signaling	a signaling pathway name
signaling.name	alternative signaling pathway name to show on the plot
color.use	the character vector defining the color of each cell group
thresh	threshold of the p-value for determining significant interaction
vertex.receiver	a numeric vector giving the index of the cell groups as targets in the first hierarchy plot
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
idents.use	a vector giving the index or the name of cell groups of interest.
top	the fraction of interactions to show
remove.isolate	whether remove the isolate nodes in the communication network
vertex.weight	The weight of vertex: either a scale value or a vector Default is a scale value being 1, indicating all vertex is plotted in the same size; Set 'vertex.weight' as a vector to plot vertex in different size; setting 'vertex.weight = NULL' will have vertex with different size that are portional to the number of cells in each cell group.
vertex.weight.max	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	the maximum vertex size for visualization

weight.scale	whether scale the edge weight
edge.weight.max	the maximum weight of edge; defualt = max(net)
edge.width.max	The maximum edge width for visualization
layout	"hierarchy", "circle", "chord" or "spatial"
pt.title	font size of the text
title.space	the space between the title and plot
vertex.label.cex	The label size of vertex in the network
	Parameters below are set for "spatial" diagram. Please also check the function 'netVisual_spatial' for more parameters.
sample.use	the sample used for visualization, which should be the element in 'object@meta$samples'.
alpha.image	the transparency of individual spots
point.size	the size of spots
	Parameters below are set for "chord" diagram. Please also check the function 'netVisual_chord_cell' for more parameters.
group	A named group labels for making multiple-group Chord diagrams. The sector names should be used as the names in the vector. The order of group controls the sector orders and if group is set as a factor, the order of levels controls the order of groups.
cell.order	a char vector defining the cell type orders (sector orders)
small.gap	Small gap between sectors.
big.gap	Gap between the different sets of sectors, which are defined in the 'group' parameter
scale	scale each sector to same width; default = FALSE; however, it is set to be TRUE when remove.isolate = TRUE
reduce	if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.
show.legend	whether show the figure legend
legend.pos.x, legend.pos.y	adjust the legend position
...	other parameters (e.g.,vertex.label.cex, vertex.label.color, alpha.edge, label.edge, edge.label.color, edge.label.cex, edge.curved, text.x,text.y) passing to 'netVisual_hierarchy1','netVisual_hierarchy2','netVisual_circle','netVisual_spatial'. NB: some parameters might be not supported

Value

an object of class "recordedplot" or ggplot

`netVisual_barplot` *Visualization of (differential) number of interactions*

Description

Visualization of (differential) number of interactions

Usage

```
netVisual_barplot(
  object,
  comparison = c(1, 2),
  measure = c("count", "weight"),
  sources.use = NULL,
  targets.use = NULL,
  invert.source = FALSE,
  invert.target = FALSE,
  signaling = NULL,
  slot.name = c("netP", "net"),
  color.use = NULL,
  title.name = NULL,
  x.lab.rot = FALSE,
  ...
)
```

Arguments

`object`	A merged CellChat object or a single CellChat object
`comparison`	a numerical vector giving the datasets for comparison in object.list; e.g., comparison = c(1,2)
`measure`	"count" or "weight". "count": comparing the number of interactions; "weight": comparing the total interaction weights (strength)
`sources.use`	a vector giving the index or the name of source cell groups
`targets.use`	a vector giving the index or the name of target cell groups.
`invert.source, invert.target`	retain the complementary set
`signaling`	a character vector giving the name of signaling networks in a single CellChat object
`slot.name`	the slot name of object. Set is to be "netP" if input signaling is a pathway name; Set is to be "net" if input signaling is a ligand-receptor pair
`color.use`	the character vector defining the color of each cell group
`title.name`	the name of the title
`x.lab.rot`	do rotation for the x-ticklabels
`...`	Parameters passing to ‘barplot_internal‘

Value

an object of ggplot

`netVisual_bubble` *Show all the significant interactions (L-R pairs) from some cell groups to other cell groups*

Description

The dot color and size represent the calculated communication probability and p-values.

Usage

```
netVisual_bubble(
  object,
  sources.use = NULL,
  targets.use = NULL,
  signaling = NULL,
  pairLR.use = NULL,
  sort.by.source = FALSE,
  sort.by.target = FALSE,
  sort.by.source.priority = TRUE,
  color.heatmap = c("Spectral", "viridis"),
  n.colors = 10,
  direction = -1,
  thresh = 0.05,
  comparison = NULL,
  group = NULL,
  remove.isolate = FALSE,
  max.dataset = NULL,
  min.dataset = NULL,
  min.quantile = 0,
  max.quantile = 1,
  line.on = TRUE,
  line.size = 0.2,
  color.text.use = TRUE,
  color.text = NULL,
  dot.size.min = NULL,
  dot.size.max = NULL,
  title.name = NULL,
  font.size = 10,
  font.size.title = 10,
  show.legend = TRUE,
  grid.on = TRUE,
  color.grid = "grey90",
  angle.x = 90,
```

```
  vjust.x = NULL,
  hjust.x = NULL,
  return.data = FALSE
)
```

Arguments

object	CellChat object
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
signaling	a character vector giving the name of signaling pathways of interest
pairLR.use	a data frame consisting of one column named either "interaction_name" or "pathway_name", defining the interactions of interest and the order of L-R on y-axis
sort.by.source, sort.by.target, sort.by.source.priority	set the order of interacting cell pairs on x-axis; please check examples for details
color.heatmap	A character string or vector indicating the colormap option to use. It can be the avaibale color palette in viridis_pal() or brewer.pal()
n.colors	number of basic colors to generate from color palette
direction	Sets the order of colors in the scale. If 1, the default colors are used. If -1, the order of colors is reversed.
thresh	threshold of the p-value for determining significant interaction
comparison	a numerical vector giving the datasets for comparison in the merged object; e.g., comparison = c(1,2)
group	a numerical vector giving the group information of different datasets; e.g., group = c(1,2,2)
remove.isolate	whether to remove the entire empty columns, i.e., communication between certain cell groups
max.dataset	a scale, keeping the communications with highest probability in max.dataset (i.e., certrain condition)
min.dataset	a scale, keeping the communications with lowest probability in min.dataset (i.e., certrain condition)
min.quantile, max.quantile	minimum and maximum quantile cutoff values for the colorbar, may specify quantile in [0,1]
line.on	whether to add vertical line when doing comparison analysis for the merged object
line.size	size of vertical line if added
color.text.use	whether to color the xtick labels according to the dataset origin when doing comparison analysis
color.text	the colors for xtick labels according to the dataset origin when doing comparison analysis
dot.size.min, dot.size.max	Size of smallest and largest points

title.name	main title of the plot
font.size, font.size.title	font size of all the text and the title name
show.legend	whether to show legend
grid.on, color.grid	whether to add grid
angle.x, vjust.x, hjust.x	parameters for adjusting the rotation of xtick labels
return.data	whether to return the data.frame for replotting

Examples

```
## Not run:
# show all the significant interactions (L-R pairs) from some cell groups (defined by 'sources.use') to other cell g
netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:11), remove.isolate = FALSE)

# show all the significant interactions (L-R pairs) associated with certain signaling pathways
netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:11), signaling = c("CCL","CXCL"))

# show all the significant interactions (L-R pairs) based on user's input (defined by `pairLR.use`; the order of L-R
pairLR.use <- extractEnrichedLR(cellchat, signaling = c("CCL","CXCL","FGF"))
netVisual_bubble(cellchat, sources.use = c(3,4), targets.use = c(5:8), pairLR.use = pairLR.use, remove.isolate = T

# set the order of interacting cell pairs on x-axis
# (1) Default: first sort cell pairs based on the appearance of sources in levels(object@idents), and then based on t
# (2) sort cell pairs based on the targets.use defined by users
netVisual_bubble(cellchat, targets.use = c("LC","Inflam. DC","cDC2","CD40LG+ TC"), pairLR.use = pairLR.use, remov
# (3) sort cell pairs based on the sources.use defined by users
netVisual_bubble(cellchat, sources.use = c("FBN1+ FIB","APOE+ FIB","Inflam. FIB"), pairLR.use = pairLR.use, remove
# (4) sort cell pairs based on the sources.use and then targets.use defined by users
netVisual_bubble(cellchat, sources.use = c("FBN1+ FIB","APOE+ FIB","Inflam. FIB"), targets.use = c("LC","Inflam. D
# (5) sort cell pairs based on the targets.use and then sources.use defined by users
netVisual_bubble(cellchat, sources.use = c("FBN1+ FIB","APOE+ FIB","Inflam. FIB"), targets.use = c("LC","Inflam. D

# show all the increased interactions in the second dataset compared to the first dataset
netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:8), remove.isolate = TRUE, max.dataset = 2)

# show all the decreased interactions in the second dataset compared to the first dataset
netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:8), remove.isolate = TRUE, max.dataset = 1)

## End(Not run)
```

---

netVisual_chord_cell *Chord diagram for visualizing cell-cell communication for a signaling pathway*

Description

Names of cell states will be displayed in this chord diagram

Usage

```
netVisual_chord_cell(
  object,
  signaling = NULL,
  net = NULL,
  slot.name = "netP",
  color.use = NULL,
  group = NULL,
  cell.order = NULL,
  sources.use = NULL,
  targets.use = NULL,
  lab.cex = 0.8,
  small.gap = 1,
  big.gap = 10,
  annotationTrackHeight = c(0.03),
  remove.isolate = FALSE,
  link.visible = TRUE,
  scale = FALSE,
  directional = 1,
  link.target.prop = TRUE,
  reduce = -1,
  transparency = 0.4,
  link.border = NA,
  title.name = NULL,
  show.legend = FALSE,
  legend.pos.x = 20,
  legend.pos.y = 20,
  nCol = NULL,
  thresh = 0.05,
  ...
)
```

Arguments

object	CellChat object
signaling	a character vector giving the name of signaling networks
net	a weighted matrix or a data frame with three columns defining the cell-cell communication network
slot.name	the slot name of object: slot.name = "net" when visualizing cell-cell communication network per each ligand-receptor pair associated with a given signaling pathway; slot.name = "netP" when visualizing cell-cell communication network at the level of signaling pathways
color.use	colors for the cell groups
group	A named group labels for making multiple-group Chord diagrams. The sector names should be used as the names in the vector. The order of group controls the sector orders and if group is set as a factor, the order of levels controls the order of groups.

cell.order	a char vector defining the cell type orders (sector orders)
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
lab.cex	font size for the text
small.gap	Small gap between sectors.
big.gap	Gap between the different sets of sectors, which are defined in the 'group' parameter
annotationTrackHeight	annotationTrack Height
remove.isolate	whether remove sectors without any links
link.visible	whether plot the link. The value is logical, if it is set to FALSE, the corresponding link will not plotted, but the space is still ocuppied. The format is a matrix with names or a data frame with three columns
scale	scale each sector to same width; default = FALSE; however, it is set to be TRUE when remove.isolate = TRUE
directional	Whether links have directions. 1 means the direction is from the first column in df to the second column, -1 is the reverse, 0 is no direction, and 2 for two directional.
link.target.prop	If the Chord diagram is directional, for each source sector, whether to draw bars that shows the proportion of target sectors.
reduce	if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.
transparency	Transparency of link colors
link.border	border for links, single scalar or a matrix with names or a data frame with three columns
title.name	title name
show.legend	whether show the figure legend
legend.pos.x, legend.pos.y	adjust the legend position
nCol	number of columns when displaying the figures
thresh	threshold of the p-value for determining significant interaction when visualizing links at the level of ligands/receptors;
...	other parameters passing to chordDiagram

Value

an object of class "recordedplot"

`netVisual_chord_cell_internal` *Chord diagram for visualizing cell-cell communication from a weighted adjacency matrix or a data frame*

Description

Names of cell states/groups will be displayed in this chord diagram

Usage

```
netVisual_chord_cell_internal(
  net,
  color.use = NULL,
  group = NULL,
  cell.order = NULL,
  sources.use = NULL,
  targets.use = NULL,
  lab.cex = 0.8,
  small.gap = 1,
  big.gap = 10,
  annotationTrackHeight = c(0.03),
  remove.isolate = FALSE,
  link.visible = TRUE,
  scale = FALSE,
  directional = 1,
  link.target.prop = TRUE,
  reduce = -1,
  transparency = 0.4,
  link.border = NA,
  title.name = NULL,
  show.legend = FALSE,
  legend.pos.x = 20,
  legend.pos.y = 20,
  ...
)
```

Arguments

`net`	a weighted matrix or a data frame with three columns defining the cell-cell communication network
`color.use`	colors for the cell groups
`group`	A named group labels for making multiple-group Chord diagrams. The sector names should be used as the names in the vector. The order of group controls the sector orders and if group is set as a factor, the order of levels controls the order of groups.

cell.order	a char vector defining the cell type orders (sector orders)
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
lab.cex	font size for the text
small.gap	Small gap between sectors.
big.gap	Gap between the different sets of sectors, which are defined in the 'group' parameter
annotationTrackHeight	annotationTrack Height
remove.isolate	whether remove sectors without any links
link.visible	whether plot the link. The value is logical, if it is set to FALSE, the corresponding link will not plotted, but the space is still ocuppied. The format is a matrix with names or a data frame with three columns
scale	scale each sector to same width; default = FALSE; however, it is set to be TRUE when remove.isolate = TRUE
directional	Whether links have directions. 1 means the direction is from the first column in df to the second column, -1 is the reverse, 0 is no direction, and 2 for two directional.
link.target.prop	If the Chord diagram is directional, for each source sector, whether to draw bars that shows the proportion of target sectors.
reduce	if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.
transparency	Transparency of link colors
link.border	border for links, single scalar or a matrix with names or a data frame with three columns
title.name	title name of the plot
show.legend	whether show the figure legend
legend.pos.x, legend.pos.y	adjust the legend position
...	other parameters passing to chordDiagram

Value

an object of class "recordedplot"

netVisual_chord_gene *Chord diagram for visualizing cell-cell communication for a set of ligands/receptors or signaling pathways*

Description

Names of ligands/receptors or signaling pathways will be displayed in this chord diagram

Usage

```
netVisual_chord_gene(
  object,
  slot.name = "net",
  color.use = NULL,
  signaling = NULL,
  pairLR.use = NULL,
  net = NULL,
  sources.use = NULL,
  targets.use = NULL,
  lab.cex = 0.8,
  small.gap = 1,
  big.gap = 10,
  annotationTrackHeight = c(0.03),
  link.visible = TRUE,
  scale = FALSE,
  directional = 1,
  link.target.prop = TRUE,
  reduce = -1,
  transparency = 0.4,
  link.border = NA,
  title.name = NULL,
  legend.pos.x = 20,
  legend.pos.y = 20,
  show.legend = TRUE,
  thresh = 0.05,
  ...
)
```

Arguments

object	CellChat object
slot.name	the slot name of object: slot.name = "net" when visualizing links at the level of ligands/receptors; slot.name = "netP" when visualizing links at the level of signaling pathways
color.use	colors for the cell groups
signaling	a character vector giving the name of signaling networks

pairLR.use	a data frame consisting of one column named either "interaction_name" or "pathway_name", defining the interactions of interest
net	A data frame consisting of the interactions of interest. net should have at least three columns: "source","target" and "interaction_name" when visualizing links at the level of ligands/receptors; "source","target" and "pathway_name" when visualizing links at the level of signaling pathway; "interaction_name" and "pathway_name" must be the matched names in CellChatDB$interaction.
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
lab.cex	font size for the text
small.gap	Small gap between sectors.
big.gap	Gap between the different sets of sectors, which are defined in the 'group' parameter
annotationTrackHeight	annotationTrack Height
link.visible	whether plot the link. The value is logical, if it is set to FALSE, the corresponding link will not plotted, but the space is still ocuppied. The format is a matrix with names or a data frame with three columns
scale	scale each sector to same width; default = FALSE; however, it is set to be TRUE when remove.isolate = TRUE
directional	Whether links have directions. 1 means the direction is from the first column in df to the second column, -1 is the reverse, 0 is no direction, and 2 for two directional.
link.target.prop	If the Chord diagram is directional, for each source sector, whether to draw bars that shows the proportion of target sectors.
reduce	if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.
transparency	Transparency of link colors
link.border	border for links, single scalar or a matrix with names or a data frame with three columns
title.name	title name of the plot
legend.pos.x, legend.pos.y	adjust the legend position
show.legend	whether show the figure legend
thresh	threshold of the p-value for determining significant interaction when visualizing links at the level of ligands/receptors;
...	other parameters to chordDiagram

Value

an object of class "recordedplot"

`netVisual_circle` *Circle plot of cell-cell communication network*

Description

The width of edges represent the strength of the communication.

Usage

```
netVisual_circle(
  net,
  color.use = NULL,
  title.name = NULL,
  sources.use = NULL,
  targets.use = NULL,
  idents.use = NULL,
  remove.isolate = FALSE,
  top = 1,
  weight.scale = FALSE,
  vertex.weight = 20,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  vertex.label.cex = 1,
  vertex.label.color = "black",
  edge.weight.max = NULL,
  edge.width.max = 8,
  alpha.edge = 0.6,
  label.edge = FALSE,
  edge.label.color = "black",
  edge.label.cex = 0.8,
  edge.curved = 0.2,
  shape = "circle",
  layout = in_circle(),
  margin = 0.2,
  vertex.size = NULL,
  arrow.width = 1,
  arrow.size = 0.2,
  text.x = 0,
  text.y = 1.5
)
```

Arguments

`net`	A weighted matrix representing the connections
`color.use`	Colors represent different cell groups
`title.name`	the name of the title

sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
idents.use	a vector giving the index or the name of cell groups of interest.
remove.isolate	whether remove the isolate nodes in the communication network
top	the fraction of interactions to show
weight.scale	whether scale the weight
vertex.weight	The weight of vertex: either a scale value or a vector
vertex.weight.max	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	the maximum vertex size for visualization
vertex.label.cex	The label size of vertex
vertex.label.color	The color of label for vertex
edge.weight.max	the maximum weight of edge; defualt = max(net)
edge.width.max	The maximum edge width for visualization
alpha.edge	the transprency of edge
label.edge	Whether or not shows the label of edges
edge.label.color	The color for single arrow
edge.label.cex	The size of label for arrows
edge.curved	Specifies whether to draw curved edges, or not. This can be a logical or a numeric vector or scalar. First the vector is replicated to have the same length as the number of edges in the graph. Then it is interpreted for each edge separately. A numeric value specifies the curvature of the edge; zero curvature means straight edges, negative values means the edge bends clockwise, positive values the opposite. TRUE means curvature 0.5, FALSE means curvature zero
shape	The shape of the vertex, currently “circle”, “square”, “csquare”, “rectangle”, “crectangle”, “vrectangle”, “pie” (see vertex.shape.pie), ‘sphere’, and “none” are supported, and only by the plot.igraph command. “none” does not draw the vertices at all, although vertex label are plotted (if given). See shapes for details about vertex shapes and vertex.shape.pie for using pie charts as vertices.
layout	The layout specification. It must be a call to a layout specification function.
margin	The amount of empty space below, over, at the left and right of the plot, it is a numeric vector of length four. Usually values between 0 and 0.5 are meaningful, but negative values are also possible, that will make the plot zoom in to a part of the graph. If it is shorter than four then it is recycled.
vertex.size	Deprecated. Use ‘vertex.weight‘
arrow.width	The width of arrows
arrow.size	the size of arrow
text.x, text.y	the x- and y-coordinates to add the text

Value

an object of class "recordedplot"

`netVisual_diffInteraction` *Circle plot showing differential cell-cell communication network between two datasets*

Description

The width of edges represent the relative number of interactions or interaction strength. Red (or blue) colored edges represent increased (or decreased) signaling in the second dataset compared to the first one.

Usage

```
netVisual_diffInteraction(
  object,
  comparison = c(1, 2),
  measure = c("count", "weight", "count.merged", "weight.merged"),
  color.use = NULL,
  color.edge = c("#b2182b", "#2166ac"),
  title.name = NULL,
  sources.use = NULL,
  targets.use = NULL,
  remove.isolate = FALSE,
  top = 1,
  weight.scale = FALSE,
  vertex.weight = 20,
  vertex.weight.max = NULL,
  vertex.size.max = 15,
  vertex.label.cex = 1,
  vertex.label.color = "black",
  edge.weight.max = NULL,
  edge.width.max = 8,
  alpha.edge = 0.6,
  label.edge = FALSE,
  edge.label.color = "black",
  edge.label.cex = 0.8,
  edge.curved = 0.2,
  shape = "circle",
  layout = in_circle(),
  margin = 0.2,
  arrow.width = 1,
  arrow.size = 0.2
)
```

Arguments

object	A merged CellChat objects
comparison	a numerical vector giving the datasets for comparison in object.list; e.g., comparison = c(1,2)
measure	"count" or "weight". "count": comparing the number of interactions; "weight": comparing the total interaction weights (strength)
color.use	Colors represent different cell groups
color.edge	Colors for indicating whether the signaling is increased (‘color.edge[1]‘) or decreased (‘color.edge[2]‘)
title.name	the name of the title
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
remove.isolate	whether remove the isolate nodes in the communication network
top	the fraction of interactions to show
weight.scale	whether scale the weight
vertex.weight	The weight of vertex: either a scale value or a vector
vertex.weight.max	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	the maximum vertex size for visualization
vertex.label.cex	The label size of vertex
vertex.label.color	The color of label for vertex
edge.weight.max	the maximum weight of edge; defualt = max(net)
edge.width.max	The maximum edge width for visualization
alpha.edge	the transprency of edge
label.edge	Whether or not shows the label of edges
edge.label.color	The color for single arrow
edge.label.cex	The size of label for arrows
edge.curved	Specifies whether to draw curved edges, or not. This can be a logical or a numeric vector or scalar. First the vector is replicated to have the same length as the number of edges in the graph. Then it is interpreted for each edge separately. A numeric value specifies the curvature of the edge; zero curvature means straight edges, negative values means the edge bends clockwise, positive values the opposite. TRUE means curvature 0.5, FALSE means curvature zero
shape	The shape of the vertex, currently “circle”, “square”, “csquare”, “rectangle”, “crectangle”, “vrectangle”, “pie” (see vertex.shape.pie), ‘sphere’, and “none” are supported, and only by the plot.igraph command. “none” does not draw the vertices at all, although vertex label are plotted (if given). See shapes for details about vertex shapes and vertex.shape.pie for using pie charts as vertices.

layout	The layout specification. It must be a call to a layout specification function.
margin	The amount of empty space below, over, at the left and right of the plot, it is a numeric vector of length four. Usually values between 0 and 0.5 are meaningful, but negative values are also possible, that will make the plot zoom in to a part of the graph. If it is shorter than four then it is recycled.
arrow.width	The width of arrows
arrow.size	the size of arrow

Value

an object of class "recordedplot"

`netVisual_embedding` *2D visualization of the learned manifold of signaling networks*

Description

2D visualization of the learned manifold of signaling networks

Usage

```
netVisual_embedding(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  color.use = NULL,
  pathway.labeled = NULL,
  top.label = 1,
  pathway.remove = NULL,
  pathway.remove.show = TRUE,
  dot.size = c(2, 6),
  label.size = 2,
  dot.alpha = 0.5,
  xlabel = "Dim 1",
  ylabel = "Dim 2",
  title = NULL,
  font.size = 10,
  font.size.title = 12,
  do.label = T,
  show.legend = T,
  show.axes = T
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
color.use	defining the color for each cell group
pathway.labeled	a char vector giving the signaling names to show when labeling each point
top.label	the fraction of signaling pathways to label
pathway.remove	a character vector defining the signaling to remove
pathway.remove.show	whether show the removed signaling names
dot.size	a range defining the size of the symbol
label.size	font size of the text
dot.alpha	transparency
xlabel	label of x-axis
ylabel	label of y-axis
title	main title of the plot
font.size	font size of the text
font.size.title	font size of the title
do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

---

netVisual_embeddingPairwise *2D visualization of the joint manifold learning of signaling networks from two datasets*

Description

2D visualization of the joint manifold learning of signaling networks from two datasets

Usage

```
netVisual_embeddingPairwise(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  comparison = NULL,
  color.use = NULL,
  point.shape = NULL,
  pathway.labeled = NULL,
  top.label = 1,
  pathway.remove = NULL,
  pathway.remove.show = TRUE,
  dot.size = c(2, 6),
  label.size = 2.5,
  dot.alpha = 0.5,
  xlabel = "Dim 1",
  ylabel = "Dim 2",
  title = NULL,
  do.label = T,
  show.legend = T,
  show.axes = T
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
comparison	a numerical vector giving the datasets for comparison. Default are all datasets when object is a merged object
color.use	defining the color for each cell group
point.shape	a numeric vector giving the point shapes. By default point.shape <- c(21, 0, 24, 23, 25, 10, 12), see available shapes at http://www.sthda.com/english/wiki/r-plot-pch-symbols-the-different-point-shapes-available-in-r
pathway.labeled	a char vector giving the signaling names to show when labeling each point
top.label	the fraction of signaling pathways to label
pathway.remove	a character vector defining the signaling to remove
pathway.remove.show	whether show the removed signaling names
dot.size	a range defining the size of the symbol
label.size	font size of the text
dot.alpha	transparency
xlabel	label of x-axis

ylabel	label of y-axis
title	main title of the plot
do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

`netVisual_embeddingPairwiseZoomIn` *Zoom into the 2D visualization of the joint manifold learning of signaling networks from two datasets*

Description

Zoom into the 2D visualization of the joint manifold learning of signaling networks from two datasets

Usage

```
netVisual_embeddingPairwiseZoomIn(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  comparison = NULL,
  color.use = NULL,
  nCol = 1,
  point.shape = NULL,
  pathway.remove = NULL,
  dot.size = c(2, 6),
  label.size = 2.8,
  dot.alpha = 0.5,
  xlabel = NULL,
  ylabel = NULL,
  do.label = T,
  show.legend = F,
  show.axes = T
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
comparison	a numerical vector giving the datasets for comparison. Default are all datasets when object is a merged object

color.use	defining the color for each cell group
nCol	number of columns in the plot
point.shape	a numeric vector giving the point shapes. By default point.shape <- c(21, 0, 24, 23, 25, 10, 12), see available shapes at http://www.sthda.com/english/wiki/r-plot-pch-symbols-the-different-point-shapes-available-in-r
pathway.remove	a character vector defining the signaling to remove
dot.size	a range defining the size of the symbol
label.size	font size of the text
dot.alpha	transparency
xlabel	label of x-axis
ylabel	label of y-axis
do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

`netVisual_embeddingZoomIn` *Zoom into the 2D visualization of the learned manifold learning of the signaling networks*

Description

Zoom into the 2D visualization of the learned manifold learning of the signaling networks

Usage

```
netVisual_embeddingZoomIn(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  color.use = NULL,
  pathway.remove = NULL,
  nCol = 1,
  dot.size = c(2, 6),
  label.size = 2.8,
  dot.alpha = 0.5,
  xlabel = NULL,
  ylabel = NULL,
  do.label = T,
  show.legend = F,
  show.axes = T
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
color.use	defining the color for each cell group
pathway.remove	a character vector defining the signaling to remove
nCol	the number of columns of the plot
dot.size	a range defining the size of the symbol
label.size	font size of the text
dot.alpha	transparency
xlabel	label of x-axis
ylabel	label of y-axis
do.label	label the each point
show.legend	whether show the legend
show.axes	whether show the axes

---

netVisual_heatmap *Visualization of network using heatmap*

Description

This heatmap can be used to 1) show differential number of interactions or interaction strength in the cell-cell communication network between two datasets; 2) the number of interactions or interaction strength in a single dataset; 3) the inferred cell-cell communication network in a single dataset, defined by 'signaling'. Please see @Details below for detailed explanations of this heatmap plot.

Usage

```
netVisual_heatmap(
  object,
  comparison = c(1, 2),
  measure = c("count", "weight"),
  signaling = NULL,
  slot.name = c("netP", "net"),
  color.use = NULL,
  color.heatmap = NULL,
  title.name = NULL,
  width = NULL,
  height = NULL,
  ylim.top = NULL,
  ylim.right = NULL,
```

```
  font.size = 8,
  font.size.title = 10,
  cluster.rows = FALSE,
  cluster.cols = FALSE,
  sources.use = NULL,
  targets.use = NULL,
  remove.isolate = FALSE,
  row.show = NULL,
  col.show = NULL
)
```

Arguments

object	A merged CellChat object or a single CellChat object
comparison	a numerical vector giving the datasets for comparison in object.list; e.g., comparison = c(1,2)
measure	"count" or "weight". "count": comparing the number of interactions; "weight": comparing the total interaction weights (strength)
signaling	a character vector giving the name of signaling networks in a single CellChat object
slot.name	the slot name of object. Set is to be "netP" if input signaling is a pathway name; Set is to be "net" if input signaling is a ligand-receptor pair
color.use	the character vector defining the color of each cell group
color.heatmap	A vector of two colors corresponding to max/min values, or a color name in brewer.pal only when the data in the heatmap do not contain negative values. By default, color.heatmap = c('#2166ac','#b2182b') when taking a merged CellChat object as input; color.heatmap = "Reds" when taking a single CellChat object as input.
title.name	the name of the title
width	width of heatmap
height	height of heatmap
ylim.top	set the range of the top barplot (e.g., ylim.top = c(0, 4))
ylim.right	set the range of the right barplot (e.g., ylim.right = c(0, 5))
font.size	fontsize in heatmap
font.size.title	font size of the title
cluster.rows	whether cluster rows
cluster.cols	whether cluster columns
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
remove.isolate	whether remove the isolate nodes in the communication network
row.show, col.show	a vector giving the index or the name of row or columns to show in the heatmap

Details

When show differential number of interactions or interaction strength in the cell-cell communication network between two datasets, the width of edges represent the relative number of interactions or interaction strength. Red (or blue) colored edges represent increased (or decreased) signaling in the second dataset compared to the first one.

The top colored bar plot represents the sum of absolute values displayed in each column of the heatmap. The right colored bar plot represents the sum of absolute values in each row.

Value

an object of ComplexHeatmap

`netVisual_hierarchy1` *Hierarchy plot of cell-cell communications sending to cell groups in vertex.receiver*

Description

The width of edges represent the strength of the communication.

Usage

```
netVisual_hierarchy1(
  net,
  vertex.receiver,
  color.use = NULL,
  title.name = NULL,
  sources.use = NULL,
  targets.use = NULL,
  remove.isolate = FALSE,
  top = 1,
  weight.scale = FALSE,
  vertex.weight = 20,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  edge.weight.max = NULL,
  edge.width.max = 8,
  alpha.edge = 0.6,
  label.dist = 2.8,
  space.v = 1.5,
  space.h = 1.6,
  shape = NULL,
  label.edge = FALSE,
  edge.curved = 0,
  margin = 0.2,
  vertex.label.cex = 0.6,
```

```
  vertex.label.color = "black",
  arrow.width = 1,
  arrow.size = 0.2,
  edge.label.color = "black",
  edge.label.cex = 0.5,
  vertex.size = NULL
)
```

Arguments

net	a weighted matrix defining the signaling network
vertex.receiver	a numeric vector giving the index of the cell groups as targets in the first hierarchy plot
color.use	the character vector defining the color of each cell group
title.name	alternative signaling pathway name to show on the plot
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
remove.isolate	whether remove the isolate nodes in the communication network
top	the fraction of interactions to show
weight.scale	whether rescale the edge weights
vertex.weight	The weight of vertex: either a scale value or a vector
vertex.weight.max	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	the maximum vertex size for visualization
edge.weight.max	the maximum weight of edge; defualt = max(net)
edge.width.max	The maximum edge width for visualization
alpha.edge	the transprency of edge
label.dist	the distance between labels and dot position
space.v	the space between different columns in the plot
space.h	the space between different rows in the plot
shape	The shape of the vertex, currently “circle”, “square”, “csquare”, “rectangle”, “crectangle”, “vrectangle”, “pie” (see vertex.shape.pie), ‘sphere’, and “none” are supported, and only by the plot.igraph command. “none” does not draw the vertices at all, although vertex label are plotted (if given). See shapes for details about vertex shapes and vertex.shape.pie for using pie charts as vertices.
label.edge	whether label edge
edge.curved	Specifies whether to draw curved edges, or not. This can be a logical or a numeric vector or scalar. First the vector is replicated to have the same length as the number of edges in the graph. Then it is interpreted for each edge separately. A numeric value specifies the curvature of the edge; zero curvature means straight edges, negative values means the edge bends clockwise, positive values the opposite. TRUE means curvature 0.5, FALSE means curvature zero

margin The amount of empty space below, over, at the left and right of the plot, it is a numeric vector of length four. Usually values between 0 and 0.5 are meaningful, but negative values are also possible, that will make the plot zoom in to a part of the graph. If it is shorter than four then it is recycled.

vertex.label.cex The label size of vertex

vertex.label.color The color of label for vertex

arrow.width The width of arrows

arrow.size the size of arrow

edge.label.color The color for single arrow

edge.label.cex The size of label for arrows

vertex.size Deprecated. Use ‘vertex.weight‘

Value

an object of class "recordedplot"

---

netVisual_hierarchy2 *Hierarchy plot of cell-cell communication sending to cell groups not in vertex.receiver*

Description

This function loads the significant interactions as a weighted matrix, and colors represent different types of cells as a structure. The width of edges represent the strength of the communication.

Usage

```
netVisual_hierarchy2(
  net,
  vertex.receiver,
  color.use = NULL,
  title.name = NULL,
  sources.use = NULL,
  targets.use = NULL,
  remove.isolate = FALSE,
  top = 1,
  weight.scale = FALSE,
  vertex.weight = 20,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  edge.weight.max = NULL,
  edge.width.max = 8,
  alpha.edge = 0.6,
```

```
  label.dist = 2.8,
  space.v = 1.5,
  space.h = 1.6,
  shape = NULL,
  label.edge = FALSE,
  edge.curved = 0,
  margin = 0.2,
  vertex.label.cex = 0.6,
  vertex.label.color = "black",
  arrow.width = 1,
  arrow.size = 0.2,
  edge.label.color = "black",
  edge.label.cex = 0.5,
  vertex.size = NULL
)
```

Arguments

net	a weighted matrix defining the signaling network
vertex.receiver	a numeric vector giving the index of the cell groups as targets in the first hierarchy plot
color.use	the character vector defining the color of each cell group
title.name	alternative signaling pathway name to show on the plot
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
remove.isolate	whether remove the isolate nodes in the communication network
top	the fraction of interactions to show
weight.scale	whether rescale the edge weights
vertex.weight	The weight of vertex: either a scale value or a vector
vertex.weight.max	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	the maximum vertex size for visualization
edge.weight.max	the maximum weight of edge; defualt = max(net)
edge.width.max	The maximum edge width for visualization
alpha.edge	the transprency of edge
label.dist	the distance between labels and dot position
space.v	the space between different columns in the plot
space.h	the space between different rows in the plot
shape	The shape of the vertex, currently “circle”, “square”, “csquare”, “rectangle”, “crectangle”, “vrectangle”, “pie” (see vertex.shape.pie), ‘sphere’, and “none” are supported, and only by the plot.igraph command. “none” does not draw the vertices at all, although vertex label are plotted (if given). See shapes for details about vertex shapes and vertex.shape.pie for using pie charts as vertices.

label.edge Whether or not shows the label of edges (number of connections between different cell types)

edge.curved Specifies whether to draw curved edges, or not. This can be a logical or a numeric vector or scalar. First the vector is replicated to have the same length as the number of edges in the graph. Then it is interpreted for each edge separately. A numeric value specifies the curvature of the edge; zero curvature means straight edges, negative values means the edge bends clockwise, positive values the opposite. TRUE means curvature 0.5, FALSE means curvature zero

margin The amount of empty space below, over, at the left and right of the plot, it is a numeric vector of length four. Usually values between 0 and 0.5 are meaningful, but negative values are also possible, that will make the plot zoom in to a part of the graph. If it is shorter than four then it is recycled.

vertex.label.cex
The label size of vertex

vertex.label.color
The color of label for vertex

arrow.width The width of arrows

arrow.size the size of arrow

edge.label.color
The color for single arrow

edge.label.cex The size of label for arrows

vertex.size Deprecated. Use ‘vertex.weight‘

Value

an object of class "recordedplot"

`netVisual_individual` *Visualize the inferred signaling network of individual L-R pairs*

Description

Visualize the inferred signaling network of individual L-R pairs

Usage

```
netVisual_individual(
  object,
  signaling,
  signaling.name = NULL,
  pairLR.use = NULL,
  color.use = NULL,
  vertex.receiver = NULL,
  sources.use = NULL,
  targets.use = NULL,
```

```
  top = 1,
  remove.isolate = FALSE,
  vertex.weight = 1,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  vertex.label.cex = 0.8,
  weight.scale = TRUE,
  edge.weight.max = NULL,
  edge.width.max = 8,
  graphics.init = TRUE,
  layout = c("circle", "hierarchy", "chord", "spatial"),
  height = 5,
  thresh = 0.05,
  alpha.image = 0.15,
  point.size = 1.5,
  group = NULL,
  cell.order = NULL,
  small.gap = 1,
  big.gap = 10,
  scale = FALSE,
  reduce = -1,
  show.legend = FALSE,
  legend.pos.x = 20,
  legend.pos.y = 20,
  nCol = NULL,
  ...
)
```

Arguments

object	CellChat object
signaling	a signaling pathway name
signaling.name	alternative signaling pathway name to show on the plot
pairLR.use	a char vector or a data frame consisting of one column named "interaction_name", defining the L-R pairs of interest
color.use	the character vector defining the color of each cell group
vertex.receiver	a numeric vector giving the index of the cell groups as targets in the first hierarchy plot
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
top	the fraction of interactions to show
remove.isolate	whether remove the isolate nodes in the communication network
vertex.weight	The weight of vertex: either a scale value or a vector. Default is a scale value being 1, indicating all vertex is plotted in the same size;

Set ‘vertex.weight‘ as a vector to plot vertex in different size; setting ‘vertex.weight = NULL‘ will have vertex with different size that are portional to the number of cells in each cell group.

`vertex.weight.max`
: the maximum weight of vertex; defualt = max(vertex.weight)

`vertex.size.max`
: the maximum vertex size for visualization

`vertex.label.cex`
: The label size of vertex in the network

`weight.scale`
: whether scale the edge weight

`edge.weight.max`
: the maximum weight of edge; defualt = max(net)

`edge.width.max`
: The maximum edge width for visualization

`graphics.init`
: whether do graphics initiation using par(...). If graphics.init=FALSE, USERS can use par() in a more fexible way

`layout`
: "hierarchy", "circle" or "chord"

`height`
: height of plot

`thresh`
: threshold of the p-value for determining significant interaction Parameters below are set for "spatial" diagram. Please also check the function ‘netVisual_spatial‘ for more parameters.

`alpha.image`
: the transparency of individual spots

`point.size`
: the size of spots

Parameters below are set for "chord" diagram. Please also check the function ‘netVisual_chord_cell‘ for more parameters.

`group`
: A named group labels for making multiple-group Chord diagrams. The sector names should be used as the names in the vector. The order of group controls the sector orders and if group is set as a factor, the order of levels controls the order of groups.

`cell.order`
: a char vector defining the cell type orders (sector orders)

`small.gap`
: Small gap between sectors.

`big.gap`
: Gap between the different sets of sectors, which are defined in the ‘group‘ parameter

`scale`
: scale each sector to same width; default = FALSE; however, it is set to be TRUE when remove.isolate = TRUE

`reduce`
: if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.

`show.legend`
: whether show the figure legend

`legend.pos.x`, `legend.pos.y`
: adjust the legend position

`nCol`
: number of columns when displaying the figures using "circle" or "chord"

`...`
: other parameters (e.g.,vertex.label.cex, vertex.label.color, alpha.edge, label.edge, edge.label.color, edge.label.cex, edge.curved, text.x, text.y) passing to ‘netVisual_hierarchy1‘,‘netVisual_hierarchy2‘,‘netVisual_circle‘. NB: some parameters might be not supported

Value

an object of class "recordedplot"

`netVisual_spatial` *Spatial plot of cell-cell communication network*

Description

Autocrine interactions are omitted on this plot. Group centroids may be not accurate for some data due to complex geometry. The width of edges represent the strength of the communication.

Usage

```
netVisual_spatial(
  net,
  coordinates,
  meta,
  sample.use = NULL,
  color.use = NULL,
  title.name = NULL,
  sources.use = NULL,
  targets.use = NULL,
  idents.use = NULL,
  remove.isolate = FALSE,
  remove.loop = TRUE,
  top = 1,
  weight.scale = FALSE,
  vertex.weight = 20,
  vertex.weight.max = NULL,
  vertex.size.max = NULL,
  vertex.label.cex = 5,
  vertex.label.color = "black",
  edge.weight.max = NULL,
  edge.width.max = 8,
  edge.curved = 0.2,
  alpha.edge = 0.6,
  arrow.angle = 5,
  arrow.size = 0.2,
  alpha.image = 0.15,
  point.size = 1.5,
  legend.size = 5
)
```

Arguments

`net`	A weighted matrix representing the connections

coordinates	a data matrix in which each row gives the spatial locations/coordinates of each cell/spot
meta	a data frame with at least two columns named 'labels' and 'samples'. 'meta$labels' is a vector giving the group label of each cell/spot. 'meta$samples' is a factor vector defining the sample labels of each dataset. The length should be the same as the number of rows in 'coordinates'.
sample.use	the sample used for visualization, which should be the element in 'meta$samples'.
color.use	Colors represent different cell groups
title.name	the name of the title
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
idents.use	a vector giving the index or the name of cell groups of interest.
remove.isolate	whether remove the isolate nodes in the communication network
remove.loop	whether remove the self-loop in the communication network. Default: TRUE
top	the fraction of interactions to show
weight.scale	whether scale the weight
vertex.weight	The weight of vertex: either a scale value or a vector
vertex.weight.max	the maximum weight of vertex; defualt = max(vertex.weight)
vertex.size.max	the maximum vertex size for visualization
vertex.label.cex	The label size of vertex
vertex.label.color	The color of label for vertex
edge.weight.max	the maximum weight of edge; defualt = max(net)
edge.width.max	The maximum edge width for visualization
edge.curved	Specifies whether to draw curved edges, or not. This can be a logical or a numeric vector or scalar. First the vector is replicated to have the same length as the number of edges in the graph. Then it is interpreted for each edge separately. A numeric value specifies the curvature of the edge; zero curvature means straight edges, negative values means the edge bends clockwise, positive values the opposite. TRUE means curvature 0.5, FALSE means curvature zero
alpha.edge	the transprency of edge
arrow.angle	The width of arrows
arrow.size	the size of arrow
alpha.image	the transparency of individual spots
point.size	the size of spots
legend.size	the size of legend

Value

an object of ggplot

nnd *compute nnd*

Description

compute nnd

Usage

```
nnd(g)
```

Arguments

g	a graph objecct

node_distance *compute the node distance matrix*

Description

compute the node distance matrix

Usage

```
node_distance(g)
```

Arguments

g	a graph objecct

normalizeData *Normalize data using a scaling factor*

Description

Normalize data using a scaling factor

Usage

```
normalizeData(data.raw, scale.factor = 10000, do.log = TRUE, do.sparse = TRUE)
```

Arguments

data.raw	input raw data
scale.factor	the scaling factor used for each cell
do.log	whether to do log transformation with pseudocount 1
do.sparse	whether to use sparse format

pieChart *Plot pie chart*

Description

Plot pie chart

Usage

```
pieChart(df, label.size = 2.5, color.use = NULL, title = "")
```

Arguments

df	a dataframe
label.size	a character
color.use	the name of the variable in CellChatDB interaction_input
title	the title of plot

plotGeneExpression *A Seurat wrapper function for plotting gene expression using violin plot, dot plot or bar plot*

Description

This function create a Seurat object from an input CellChat object, and then plot gene expression distribution using a modified violin plot or dot plot based on Seurat's function or a bar plot. Please check StackedVlnPlot,dotPlot and barPlotfor detailed description of the arguments.

Usage

```
plotGeneExpression(
  object,
  features = NULL,
  signaling = NULL,
  enriched.only = TRUE,
  type = c("violin", "dot", "bar"),
  color.use = NULL,
  group.by = NULL,
  ...
)
```

Arguments

object	CellChat object
features	Features to plot gene expression
signaling	a char vector containing signaling pathway names for searching
enriched.only	whether only return the identified enriched signaling genes in the database. Default = TRUE, returning the significantly enriched signaling interactions
type	violin plot or dot plot
color.use	defining the color for each cell group
group.by	Name of one metadata columns to group (color) cells. Default is the defined cell groups in CellChat object
...	other arguments passing to either VlnPlot or DotPlot from Seurat package

Details

USER can extract the signaling genes related to the inferred L-R pairs or signaling pathway using extractEnrichedLR, and then plot gene expression using Seurat package.

PPI.human *Human Protein-Protein interactions*

Description

An adjacency matrix of the high confidence known experimental interactions between human proteins on STRINGdb.

Usage

```
PPI.human
```

Format

A square matrix where A_ij=1 if protein i interacts with protein j

Source

http://www.string-db.org/

PPI.mouse *Mouse Protein-Protein interactions*

Description

An adjacency matrix of the high confidence known experimental interactions between mouse proteins on STRINGdb.

Usage

```
PPI.mouse
```

Format

A square matrix where A_ij=1 if protein i interacts with protein j

Source

http://www.string-db.org/

preProcMultiomics *Preprocessing multi-omics data and preparing the L-R database*

Description

Preprocessing multi-omics data and preparing the L-R database

Usage

```
preProcMultiomics(data.list, db, cutoff = 0.5, do.sparse = TRUE)
```

Arguments

data.list	a list consisting of multi-omics data (e.g., RNA & ADT)
db	one of the CellChatDB databases: CellChatDB.human, CellChatDB.mouse, CellChatDB.zebrafish
cutoff	the cutoff value of low protein expression
do.sparse	whether to use sparse format

rankNet *Rank signaling networks based on the information flow or the number of interactions*

Description

This function can also be used to rank signaling from certain cell groups to other cell groups

Usage

```
rankNet(
  object,
  slot.name = "netP",
  measure = c("weight", "count"),
  mode = c("comparison", "single"),
  comparison = c(1, 2),
  color.use = NULL,
  stacked = FALSE,
  sources.use = NULL,
  targets.use = NULL,
  signaling = NULL,
  pairLR = NULL,
  signaling.type = NULL,
  do.stat = FALSE,
  paired.test = TRUE,
  cutoff.pvalue = 0.05,
  tol = 0.05,
  thresh = 0.05,
  show.raw = FALSE,
  return.data = FALSE,
  x.rotation = 90,
  title = NULL,
  bar.w = 0.75,
  font.size = 8,
  do.flip = TRUE,
  x.angle = NULL,
  y.angle = 0,
  x.hjust = 1,
  y.hjust = 1,
  axis.gap = FALSE,
  ylim = NULL,
  segments = NULL,
  tick_width = NULL,
  rel_heights = c(0.9, 0, 0.1)
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
measure	"weight" or "count". "weight": comparing the total interaction weights (strength); "count": comparing the number of interactions;
mode	"single","comparison"
comparison	a numerical vector giving the datasets for comparison; a single value means ranking for only one dataset and two values means ranking comparison for two datasets
color.use	defining the color for each cell group
stacked	whether plot the stacked bar plot
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
signaling	a vector giving the signaling pathway to show
pairLR	a vector giving the names of L-R pairs to show (e.g, pairLR = c("IL1A_IL1R1_IL1RAP","IL1B_IL1R1_IL
signaling.type	a char giving the types of signaling from the three categories c("Secreted Signaling", "ECM-Receptor", "Cell-Cell Contact")
do.stat	whether do a Wilcoxon test to determine whether there is significant difference between two datasets. Default = FALSE
paired.test	a logical indicating whether you want a paired test. Paired test is applicable to compare two datasets with the same cellular compositions.
cutoff.pvalue	the cutoff of pvalue when doing Wilcoxon test; Default = 0.05
tol	a tolerance when considering the relative contribution being equal between two datasets. contribution.relative between 1-tol and 1+tol will be considered as equal contribution
thresh	threshold of the p-value for determining significant interaction
show.raw	whether show the raw information flow. Default = FALSE, showing the scaled information flow to provide compariable data scale; When stacked = TRUE, use raw information flow by default.
return.data	whether return the data.frame consisting of the calculated information flow of each signaling pathway or L-R pair
x.rotation	rotation of x-labels
title	main title of the plot
bar.w	the width of bar plot
font.size	font size
do.flip	whether flip the x-y axis
x.angle, y.angle, x.hjust, y.hjust	parameters for rotating and spacing axis labels
axis.gap	whetehr making gaps in y-axes

ylim, segments, tick_width, rel_heights
: parameters in the function gg.gap when making gaps in y-axes e.g., ylim = c(0, 35), segments = list(c(11, 14),c(16, 28)), tick_width = c(5,2,5), rel_heights = c(0.8,0,0.1,0,0.1) https://tobiasbusch.xyz/an-r-package-for-everything-ep2-gaps

---

rankNetPairwise *Rank ligand-receptor interactions for any pair of two cell groups*

Description

Rank ligand-receptor interactions for any pair of two cell groups

Usage

```
rankNetPairwise(object, LR.use = NULL)
```

Arguments

object	CellChat object
LR.use	ligand-receptor interactions used in inferring communication network

---

rankSimilarity *Rank the similarity of the shared signaling pathways based on their joint manifold learning*

Description

Rank the similarity of the shared signaling pathways based on their joint manifold learning

Usage

```
rankSimilarity(
  object,
  slot.name = "netP",
  type = c("functional", "structural"),
  comparison1 = NULL,
  comparison2 = c(1, 2),
  x.rotation = 90,
  title = NULL,
  color.use = NULL,
  bar.w = NULL,
  font.size = 8
)
```

Arguments

object	CellChat object
slot.name	the slot name of object that is used to compute centrality measures of signaling networks
type	"functional","structural"
comparison1	a numerical vector giving the datasets for comparison. This should be the same as 'comparison' in 'computeNetSimilarityPairwise'
comparison2	a numerical vector with two elements giving the datasets for comparison. If there are more than 2 datasets defined in 'comparison1', 'comparison2' can be defined to indicate which two datasets used for computing the distance. e.g., comparison2 = c(1,3) indicates the first and third datasets defined in 'comparison1' will be used for comparison.
x.rotation	rotation of x-labels
title	main title of the plot
color.use	defining the color
bar.w	the width of bar plot
font.size	font size

---

runCellChatApp *Generate a Shiny App for interactive exploration of CellChat's outputs*

Description

Generate a Shiny App for interactive exploration of CellChat's outputs

Usage

```
runCellChatApp(object, ...)
```

Arguments

object	CellChat object
...	Other parameters of 'shinyApp' function from shiny R package

Value

A Shiny app object on the basis of one CellChat object

runPCA *Dimension reduction using PCA*

Description

Dimension reduction using PCA

Usage

```
runPCA(data.use, do.fast = T, dimPC = 50, seed.use = 42, weight.by.var = T)
```

Arguments

data.use	input data (samples in rows, features in columns)
do.fast	whether do fast PCA
dimPC	the number of components to keep
seed.use	set a seed
weight.by.var	whether use weighted pc.scores

runUMAP *Run UMAP*

Description

Run UMAP

Usage

```
runUMAP(
  data.use,
  n_neighbors = 30L,
  n_components = 2L,
  metric = "correlation",
  n_epochs = NULL,
  learning_rate = 1,
  min_dist = 0.3,
  spread = 1,
  set_op_mix_ratio = 1,
  local_connectivity = 1L,
  repulsion_strength = 1,
  negative_sample_rate = 5,
  a = NULL,
  b = NULL,
  seed.use = 42L,
```

```
  metric_kwds = NULL,
  angular_rp_forest = FALSE,
  verbose = FALSE
)
```

Arguments

data.use	input data matrix
n_neighbors	This determines the number of neighboring points used in local approximations of manifold structure. Larger values will result in more global structure being preserved at the loss of detailed local structure. In general this parameter should often be in the range 5 to 50.
n_components	The dimension of the space to embed into.
metric	This determines the choice of metric used to measure distance in the input space.
n_epochs	the number of training epochs to be used in optimizing the low dimensional embedding. Larger values result in more accurate embeddings. If NULL is specified, a value will be selected based on the size of the input dataset (200 for large datasets, 500 for small).
learning_rate	The initial learning rate for the embedding optimization.
min_dist	This controls how tightly the embedding is allowed compress points together. Larger values ensure embedded points are moreevenly distributed, while smaller values allow the algorithm to optimise more accurately with regard to local structure. Sensible values are in the range 0.001 to 0.5.
spread	he effective scale of embedded points. In combination with min.dist this determines how clustered/clumped the embedded points are.
set_op_mix_ratio	Interpolate between (fuzzy) union and intersection as the set operation used to combine local fuzzy simplicial sets to obtain a global fuzzy simplicial sets.
local_connectivity	The local connectivity required - i.e. the number of nearest neighbors that should be assumed to be connected at a local level. The higher this value the more connected the manifold becomes locally. In practice this should be not more than the local intrinsic dimension of the manifold.
repulsion_strength	Weighting applied to negative samples in low dimensional embedding optimization. Values higher than one will result in greater weight being given to negative samples.
negative_sample_rate	The number of negative samples to select per positive sample in the optimization process. Increasing this value will result in greater repulsive force being applied, greater optimization cost, but slightly more accuracy.
a	More specific parameters controlling the embedding. If NULL, these values are set automatically as determined by min. dist and spread.
b	More specific parameters controlling the embedding. If NULL, these values are set automatically as determined by min. dist and spread.

seed.use	Set a random seed. By default, sets the seed to 42.
metric_kwds, angular_rp_forest, verbose	other parameters used in UMAP

---

scaleData *Scale the data*

Description

Scale the data

Usage

```
scaleData(data.use, do.center = T)
```

Arguments

data.use	input data
do.center	whether center the values

---

scaleMat *Scale a data matrix*

Description

Scale a data matrix

Usage

```
scaleMat(x, scale, na.rm = TRUE)
```

Arguments

x	data matrix
scale	the method to scale the data
na.rm	whether remove na

scPalette *Generate colors from a customed color palette*

Description

Generate colors from a customed color palette

Usage

```
scPalette(n)
```

Arguments

n	number of colors

Value

A color palette for plotting

searchPair *Subset the ligand-receptor interactions for given specific signals in CellChatDB*

Description

Subset the ligand-receptor interactions for given specific signals in CellChatDB

Usage

```
searchPair(
  signaling = c(),
  pairLR.use,
  key = c("pathway_name", "ligand"),
  matching.exact = FALSE,
  pair.only = TRUE
)
```

Arguments

signaling	a character vector
pairLR.use	a dataframe containing ligand-receptor interactions
key	the keyword to match
matching.exact	whether perform exact matching
pair.only	whether only return ligand-receptor pairs without cofactors

`selectK` *Select the number of the patterns for running 'identifyCommunicationPatterns'*

Description

We infer the number of patterns based on two metrics that have been implemented in the NMF R package, including Cophenetic and Silhouette. Both metrics measure the stability for a particular number of patterns based on a hierarchical clustering of the consensus matrix. For a range of the number of patterns, a suitable number of patterns is the one at which Cophenetic and Silhouette values begin to drop suddenly.

Usage

```
selectK(
  object,
  slot.name = "netP",
  pattern = c("outgoing", "incoming"),
  title.name = NULL,
  do.facet = TRUE,
  k.range = seq(2, 10),
  nrun = 30,
  seed.use = 10
)
```

Arguments

`object`	CellChat object
`slot.name`	the slot name of object that is used to compute centrality measures of signaling networks
`pattern`	"outgoing" or "incoming"
`title.name`	title of plot
`do.facet`	whether use facet plot showing the two measures
`k.range`	a range of the number of patterns
`nrun`	number of runs when performing NMF
`seed.use`	seed when performing NMF

Value

a ggplot object

setIdent *Set the default identity of cells*

Description

Set the default identity of cells

Usage

```
setIdent(object, ident.use = NULL, levels = NULL, display.warning = TRUE)
```

Arguments

object	CellChat object
ident.use	the name of the variable in object.meta;
levels	set the levels of factor
display.warning	whether display the warning message

show,CellChat-method *show method for CellChat*

Description

show method for CellChat

Usage

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

Arguments

object	object
CellChat	object
show	show the object

showDatabaseCategory *Show the description of CellChatDB databse*

Description

Show the description of CellChatDB databse

Usage

```
showDatabaseCategory(CellChatDB, nrow = 1)
```

Arguments

CellChatDB	CellChatDB databse
nrow	the number of rows in the plot

sketchData *Downsampling single cell data using geometric sketching algorithm*

Description

USERs need to install the python package 'pip install geosketch' (https://github.com/brianhie/geosketch)

Usage

```
sketchData(object, percent, idents = NULL, do.PCA = TRUE, dimPC = 30)
```

Arguments

object	A data matrix (should have row names; samples in rows, features in columns) or a Seurat object. When object is a PCA or UMAP space, please set 'do.PCA = FALSE' When object is a data matrix (cells in rows and genes in columns), it is better to use the highly variable genes. PCA will be done on this input data matrix.
percent	the percent of data to sketch
idents	A vector of identity classes to keep for sketching
do.PCA	whether doing PCA on the input data
dimPC	the number of components to use

Value

A vector of cell names to use for downsampling

`smoothData` *Smooth the gene expression data*

Description

A diffusion process is used to smooth genes' expression values based on their neighbors' defined in a high-confidence experimentally validated protein-protein network.

Usage

```
smoothData(
  object,
  method = c("netSmooth"),
  adj = NULL,
  alpha = 0.5,
  normalizeAdjMatrix = c("rows", "columns")
)
```

Arguments

`object`	CellChat object
`method`	When method = "netSmooth", smoothing a gene's expression values based on its neighbors defined in a high-confidence experimentally validated protein-protein network.
`adj`	adjacency matrix of protein-protein interaction network to use
`alpha`	numeric in [0,1] alpha = 0: no smoothing; a larger value alpha results in increasing levels of smoothing.
`normalizeAdjMatrix`	how to normalize the adjacency matrix possible values are 'rows' (in-degree) and 'columns' (out-degree)

Details

This function is useful when analyzing single-cell data with shallow sequencing depth because the projection reduces the dropout effects of signaling genes, in particular for possible zero expression of subunits of ligands/receptors

Value

a smoothed gene expression matrix

`spatialDimPlot` *Visualize spatial cell groups*

Description

This function takes a CellChat object as input, and then plot cell groups of interest.

Usage

```
spatialDimPlot(
  object,
  color.use = NULL,
  group.by = NULL,
  sample.use = NULL,
  sources.use = NULL,
  targets.use = NULL,
  idents.use = NULL,
  alpha = 1,
  shape.by = 16,
  title.name = NULL,
  point.size = 2.4,
  legend.size = 5,
  legend.text.size = 8,
  legend.position = "right",
  ncol = 1,
  byrow = FALSE
)
```

Arguments

`object`	cellchat object
`color.use`	defining the color for each cell group
`group.by`	Name of one metadata columns to group (color) cells. Default is the defined cell groups in CellChat object
`sample.use`	the sample name used for visualization, which should be the element in 'object@meta$samples'.
`sources.use`	a vector giving the index or the name of source cell groups
`targets.use`	a vector giving the index or the name of target cell groups
`idents.use`	a vector giving the index or the name of cell groups of interest
`alpha`	the transparency of individual spot
`shape.by`	the shape of individual spot
`title.name`	title name
`point.size`	the size of spots
`legend.size`	the size of legend

legend.text.size	the text size on the legend
legend.position	legend position
ncol	number of columns of the legend text
byrow	arrange the legend text byrow or not

`spatialFeaturePlot` *A spatial feature plots*

Description

This function takes a CellChat object as input, and then plot gene expression distribution over spots/cells on the image.

Usage

```
spatialFeaturePlot(
  object,
  features = NULL,
  signaling = NULL,
  pairLR.use = NULL,
  sample.use = NULL,
  enriched.only = TRUE,
  thresh = 0.05,
  do.group = TRUE,
  color.heatmap = "Spectral",
  n.colors = 8,
  direction = -1,
  do.binary = FALSE,
  cutoff = NULL,
  color.use = NULL,
  alpha = 1,
  point.size = 0.8,
  legend.size = 3,
  legend.text.size = 8,
  shape.by = 16,
  ncol = NULL,
  show.legend = TRUE,
  show.legend.combined = FALSE
)
```

Arguments

object	cellchat object
features	a char vector containing features to visualize. ‘features‘ can be genes or column names of ‘object@meta‘.

signaling	signalling names to visualize
pairLR.use	a data frame consisting of one column named "interaction_name", defining the L-R pairs of interest
sample.use	the sample used for visualization, which should be the element in 'object@meta$samples'.
enriched.only	whether only return the identified enriched signaling genes in the database. Default = TRUE, returning the significantly enriched signaling interactions
thresh	threshold of the p-value for determining significant interaction when visualizing links at the level of ligands/receptors;
do.group	set 'do.group = TRUE' when only showing enriched signaling based on cell group-level communication; set 'do.group = FALSE' when only showing enriched signaling based on individual cell-level communication
color.heatmap	A character string or vector indicating the colormap option to use. It can be the avaibale color palette in brewer.pal() or viridis_pal() (e.g., "Spectral","viridis")
n.colors, direction	n.colors: number of basic colors to generate from color palette; direction: Sets the order of colors in the scale. If 1, the default colors are used. If -1, the order of colors is reversed.
do.binary, cutoff	whether binarizing the expression using a given cutoff
color.use	defining the color for cells/spots expressing ligand only, expressing receptor only, expressing both ligand & receptor and cells/spots without expression of given ligands and receptors
alpha	the transparency of individual spot
point.size	the size of cell slot
legend.size	the size of legend
legend.text.size	the text size on the legend
shape.by	the shape of individual spot
ncol	number of columns if plotting multiple plots
show.legend	whether show each figure legend
show.legend.combined	whether show the figure legend for the last plot

---

StackedVlnPlot *Stacked Violin plot*

Description

Stacked Violin plot

Usage

```
StackedVlnPlot(
  object,
  features,
  idents = NULL,
  split.by = NULL,
  color.use = NULL,
  colors.ggplot = FALSE,
  angle.x = 90,
  vjust.x = NULL,
  hjust.x = NULL,
  show.text.y = TRUE,
  line.size = NULL,
  pt.size = 0,
  plot.margin = margin(0, 0, 0, 0, "cm"),
  ...
)
```

Arguments

object	seurat object
features	Features to plot (gene expression, metrics)
idents	Which classes to include in the plot (default is all)
split.by	Name of a metadata column to split plot by;
color.use	defining the color for each cell group
colors.ggplot	whether use ggplot color scheme; default: colors.ggplot = FALSE
angle.x	angle for x-axis text rotation
vjust.x	adjust x axis text
hjust.x	adjust x axis text
show.text.y	whther show y-axis text
line.size	line width in the violin plot
pt.size	size of the dots
plot.margin	adjust the white space between each plot
...	Extra parameters passed to VlnPlot from Seurat package

Value

ggplot2 object

subsetCellChat *Subset CellChat object using a portion of cells*

Description

Subset CellChat object using a portion of cells

Usage

```
subsetCellChat(
  object,
  cells.use = NULL,
  idents.use = NULL,
  group.by = NULL,
  invert = FALSE,
  thresh = 0.05
)
```

Arguments

object	A CellChat object (either an object from a single dataset or a merged objects from multiple datasets)
cells.use	a char vector giving the cell barcodes to subset. If cells.use = NULL, USER must define 'idents.use'
idents.use	a subset of cell groups used for analysis
group.by	cell group information; default is 'object@idents'; otherwise it should be one of the column names of the meta slot
invert	whether invert the idents.use
thresh	threshold of the p-value for determining significant interaction. A parameter as an input of the function 'computeCommunProbPathway'

subsetCommunication *Subset the inferred cell-cell communications of interest*

Description

NB: If all arguments are NULL, it returns a data frame consisting of all the inferred cell-cell communications

Usage

```
subsetCommunication(
  object = NULL,
  net = NULL,
  slot.name = "net",
  sources.use = NULL,
  targets.use = NULL,
  signaling = NULL,
  pairLR.use = NULL,
  thresh = 0.05,
  datasets = NULL,
  ligand.pvalues = NULL,
  ligand.logFC = NULL,
  ligand.pct.1 = NULL,
  ligand.pct.2 = NULL,
  receptor.pvalues = NULL,
  receptor.logFC = NULL,
  receptor.pct.1 = NULL,
  receptor.pct.2 = NULL
)
```

Arguments

object	CellChat object
net	Alternative input is a data frame with at least with three columns defining the cell-cell communication network ("source","target","interaction_name")
slot.name	the slot name of object: slot.name = "net" when extracting the inferred communications at the level of ligands/receptors; slot.name = "netP" when extracting the inferred communications at the level of signaling pathways
sources.use	a vector giving the index or the name of source cell groups
targets.use	a vector giving the index or the name of target cell groups.
signaling	a character vector giving the name of signaling pathways of interest
pairLR.use	a data frame consisting of one column named either "interaction_name" or "pathway_name", defining the interactions of interest
thresh	threshold of the p-value for determining significant interaction
datasets	select the inferred cell-cell communications from a particular 'datasets' when inputing a data frame 'net'
ligand.pvalues, ligand.logFC, ligand.pct.1, ligand.pct.2	set threshold for ligand genes ligand.pvalues: threshold for pvalues in the differential expression gene analysis (DEG) ligand.logFC: threshold for logFoldChange in the DEG analysis; When ligand.logFC > 0, keep upgulated genes; otherwise, kepp downregulated genes ligand.pct.1: threshold for the percent of expressed genes in the defined 'positive' cell group. keep genes with percent greater than ligand.pct.1

ligand.pct.2: threshold for the percent of expressed genes in the cells except for the defined 'positive' cell group

`receptor.pvalues`, `receptor.logFC`, `receptor.pct.1`, `receptor.pct.2`
set threshold for receptor genes

Value

If input object is created from a single dataset, a data frame of the inferred cell-cell communications of interest, consisting of source, target, interaction_name, pathway_name, prob and other information

If input object is a merged object from multiple datasets, it will return a list and each element is a data frame for one dataset

Examples

```
## Not run:
# access all the inferred cell-cell communications
df.net <- subsetCommunication(cellchat)

# access all the inferred cell-cell communications at the level of signaling pathways
df.net <- subsetCommunication(cellchat, slot.name = "netP")

# Subset to certain cells with sources.use and targets.use
df.net <- subsetCommunication(cellchat, sources.use = c(1,2), targets.use = c(4,5))

# Subset to certain signaling, e.g., WNT and TGFb
df.net <- subsetCommunication(cellchat, signaling = c("WNT", "TGFb"))

## End(Not run)
```

---

subsetCommunication_internal

Subset the inferred cell-cell communications of interest

Description

NB: If all arguments are NULL, it returns a data frame consisting of all the inferred cell-cell communications

Usage

```
subsetCommunication_internal(
  net,
  LR,
  cells.level,
  slot.name = "net",
  sources.use = NULL,
```

```
  targets.use = NULL,
  signaling = NULL,
  pairLR.use = NULL,
  thresh = 0.05,
  datasets = NULL,
  ligand.pvalues = NULL,
  ligand.logFC = NULL,
  ligand.pct.1 = NULL,
  ligand.pct.2 = NULL,
  receptor.pvalues = NULL,
  receptor.logFC = NULL,
  receptor.pct.1 = NULL,
  receptor.pct.2 = NULL
)
```

Arguments

`net`, `LR`, `cells.level`
: net is object@net or a data frame; LR: object@LR$LRsig; cells.level: levels(object@idents)

`slot.name`
: the slot name of object: slot.name = "net" when extracting the inferred communications at the level of ligands/receptors; slot.name = "netP" when extracting the inferred communications at the level of signaling pathways

`sources.use`
: a vector giving the index or the name of source cell groups

`targets.use`
: a vector giving the index or the name of target cell groups.

`signaling`
: a character vector giving the name of signaling pathways of interest

`pairLR.use`
: a data frame consisting of one column named either "interaction_name" or "pathway_name", defining the interactions of interest

`thresh`
: threshold of the p-value for determining significant interaction

`datasets`
: select the inferred cell-cell communications from a particular ‘datasets‘ when inputing a data frame ‘net‘

`ligand.pvalues`, `ligand.logFC`, `ligand.pct.1`, `ligand.pct.2`
: set threshold for ligand genes

  ligand.pvalues: threshold for pvalues in the differential expression gene analysis (DEG)

  ligand.logFC: threshold for logFoldChange in the DEG analysis; When ligand.logFC > 0, keep upgulated genes; otherwise, kepp downregulated genes

  ligand.pct.1: threshold for the percent of expressed genes in the defined ’positive’ cell group. keep genes with percent greater than ligand.pct.1

  ligand.pct.2: threshold for the percent of expressed genes in the cells except for the defined ’positive’ cell group

`receptor.pvalues`, `receptor.logFC`, `receptor.pct.1`, `receptor.pct.2`
: set threshold for receptor genes

Value

A data frame of the inferred cell-cell communications of interest, consisting of source, target, interaction_name, pathway_name, prob and other information

---

subsetData *Subset the expression data of signaling genes for saving computation cost*

Description

Subset the expression data of signaling genes for saving computation cost

Usage

```
subsetData(object, features = NULL)
```

Arguments

object	CellChat object
features	default = NULL: subset the expression data of signaling genes in CellChatDB.use

Value

An updated CellChat object by assigning a subset of the data into the slot ‘data.signaling‘

---

subsetDB *Subset CellChatDB databse by only including interactions of interest*

Description

Subset CellChatDB databse by only including interactions of interest

Usage

```
subsetDB(CellChatDB, search = c(), key = "annotation", non_protein = FALSE)
```

Arguments

CellChatDB	CellChatDB databse
search	a character vector, which is a subset of c("Secreted Signaling","ECM-Receptor","Cell-Cell Contact","Non-protein Signaling"); Setting search = NULL & non_protein = FALSE will return all signaling except for "Non-protein Signaling". When ‘key‘ is a vector, the ‘search‘ should be a list with the size being ‘length(key)‘, where each element is a character vector.

key	a character vector and each element should be one of the column names of the interaction_input from CellChatDB.
non_protein	whether to use the non-protein signaling for CellChat analysis. By default, non_protein = FALSE because most of non-protein signaling are the special synaptic signaling interactions that can only be used when inferring neuron-neuron communication.

`thresholdedMean` *Compute the average expression per cell group when the percent of expressing cells per cell group larger than a threshold*

Description

Compute the average expression per cell group when the percent of expressing cells per cell group larger than a threshold

Usage

```
thresholdedMean(x, trim = 0.1, na.rm = TRUE)
```

Arguments

x	a numeric vector
trim	the percent of expressing cells per cell group to be considered as zero
na.rm	whether remove na

`triMean` *Compute the Tukey's trimean*

Description

Compute the Tukey's trimean

Usage

```
triMean(x, na.rm = TRUE)
```

Arguments

x	a numeric vector
na.rm	whether remove na

updateCCC_score *Update the cell-cell communication array from a customized cell-cell-communication scores between different cell groups*

Description

Users may also check the ‘updateCellChatDB‘ function for integrating other resources or utilizing a custom database

Usage

```
updateCCC_score(object, net)
```

Arguments

object	CellChat object
net	a data frame with at least five columns named as ‘source‘,‘target‘,‘ligand‘,‘receptor‘ and ‘score‘, which defines the customized cell-cell-communication scores between different cell groups. a p-value column named ‘pval‘, and additional columns named ‘interaction_name‘ and ‘interaction_name_2‘ can be also provided.

Value

a CellChat object with updated slot ‘net‘ and slot ‘DB‘ if db is not NULL.

updateCellChat *Update a single CellChat object*

Description

Update a single previously calculated CellChat object for spatial transcriptomics data analysis (version < 2.1.0)

Usage

```
updateCellChat(object)
```

Arguments

object	CellChat object

Details

Update a single previously calculated CellChat object (version < 1.6.0)

version < 0.5.0: ‘object@var.features‘ is now ‘object@var.features$features‘; ‘object@net$sum‘ is now ‘object@net$weight‘ if ‘aggregateNet‘ has been run.

version 1.6.0: a ‘object@images‘ slot is added and ‘datatype‘ is added in ‘object@options$datatype‘

version 2.1.0: a column named ‘slices‘ is added in ‘meta‘ data for spatial transcriptomics data analysis.

version 2.1.1: ‘images$scale.factors‘ is changed to ‘images$spatial.factors‘ for spatial transcriptomics data analysis.

version 2.1.2: the column ‘slices‘ in ‘object@meta‘ is renamed as ‘samples‘ in order to identify consistent signaling across samples for cell-cell communication analysis.

version 2.1.3: the slot ‘object@data.project‘ is renamed as ‘object@data.smooth‘.

Value

a updated CellChat object

`updateCellChatDB` *Update CellChatDB by integrating new L-R pairs from other resources or adding more information*

Description

Update CellChatDB by integrating new L-R pairs from other resources or adding more information

Usage

```
updateCellChatDB(
  db,
  gene_info = NULL,
  other_info = NULL,
  gene_info_columnNew = NULL,
  trim.pathway = FALSE,
  merged = FALSE,
  species_target = NULL
)
```

Arguments

db	a data frame of the customized ligand-receptor database with at least two columns named as ‘ligand‘ and ‘receptor‘. We highly suggest users to provide a column of pathway information named ‘pathway_name‘ associated with each L-R pair. Other optional columns include ‘interaction_name‘ and ‘interaction_name_2‘. The default columns of CellChatDB can be checked via ‘colnames(CellChatDB.human$interaction)‘.

gene_info	a data frame with at least one column named as 'Symbol'. "When setting gene_info = NULL, the input 'species_target' should be provided: either 'human' or 'mouse'.
other_info	a list consisting of other information including a dataframe named as 'complex' and a dataframe named as 'cofactor'. This additional information is not necessary. If other_info is provided, the 'complex' and 'cofactor' are dataframes with defined rownames.
gene_info_columnNew	a data frame with at least two columns named as 'Symbol' and 'AntibodyName', which will add a new column named 'AntibodyName' into 'db$geneInfo'.
trim.pathway	whether to delete the interactions with missing pathway names when the column 'pathway_name' is provided in 'db'.
merged	whether merging the input database with the existing CellChatDB. setting merged = TRUE, the input 'species_target' should be provided: either 'human' or 'mouse'.
species_target	the target species for output: either 'human' or 'mouse'.

Value

a list consisting of the customized L-R database for further CellChat analysis

Examples

```
## Not run:
# integrating new L-R pairs from other resources or utilizing a custom database `db.user`
db.new <- updateCellChatDB(db = db.user, gene_info = gene_info)
db.new <- updateCellChatDB(db = db.user, gene_info = NULL, species_target = "human")
# Alternatively, users can integrate the customized L-R pairs into the built-in CellChatDB
db.new <- updateCellChatDB(db = db.user, merged = TRUE, species_target = "human")
# Add new columns (e.g., AntibodyName) into gene_info
db.new.human <- updateCellChatDB(db = CellChatDB.human$interaction, gene_info = CellChatDB.human$geneInfo, other_

# Users can now use this new database in CellChat analysis
cellchat@DB <- db.new

## End(Not run)
```

---

updateClusterLabels *Update and re-order the cell group names after running 'computeCommunProb'*

Description

Update and re-order the cell group names after running 'computeCommunProb'

Usage

```
updateClusterLabels(
  object,
  old.cluster.name = NULL,
  new.cluster.name = NULL,
  new.order = NULL,
  new.cluster.metaname = "new.labels"
)
```

Arguments

object	CellChat object
old.cluster.name	A vector defining old cell group labels in 'object@idents'; Default = NULL, which will use 'levels(object@idents)'
new.cluster.name	A vector defining new cell group labels to rename
new.order	reset order of cell group labels
new.cluster.metaname	assign a name of the new labels, which will be the column name of new labels in 'object@meta'

Value

An updated CellChat object

Index