Package: cytosignal (via r-universe)

June 27, 2024

Title What the Package Does (One Line, Title Case)

Version 0.2.0

Type Package

Description What the package does (one paragraph).

LinkingTo Rcpp, RcppArmadillo, RcppProgress

Depends methods, R (>= 4.0), stats, utils

Imports dbscan, RANN, RTriangle, ggplot2, ggrepel, glmnet, Matrix, plyr, Rcpp, RcppArmadillo, rlang

Suggests cowplot, httr, jsonlite, patchwork, scattermore, stringi, plot3D, png, pheatmap, gprofiler2, knitr

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

VignetteBuilder knitr

Remotes xzhoulab/SPARK

Repository https://welch-lab.r-universe.dev

RemoteUrl https://github.com/welch-lab/cytosignal

RemoteRef HEAD

RemoteSha bc2573dfe78464adbeb1c17f1dfacfc54ac1de71

Contents

`.inferVeloLR.matrix_like` *#' Compute the LR velo for specific ligand-receptor imputation obj pairs #' #' @param object A Cytosignal object #' @param lig.slot The ligand slot to use #' @param recep.slot The receptor slot to use #' @param intr.db.name The intr database name to use #' @param nn.use The neighbor index as niche #' #' @return A Cytosignal object #' @export #' inferVeloLR <- function(object, ...) UseMethod(generic = 'inferVeloLR', object = object)*

Description

#' Compute the LR velo for specific ligand-receptor imputation obj pairs #' #' @param object A Cytosignal object #' @param lig.slot The ligand slot to use #' @param recep.slot The receptor slot to use #' @param intr.db.name The intr database name to use #' @param nn.use The neighbor index as niche #' #' @return A Cytosignal object #' @export #' inferVeloLR <- function(object, ...) UseMethod(generic = 'inferVeloLR', object = object)

Usage

```
.inferVeloLR.matrix_like(
  dge.lig,
  dge.recep,
  dge.lig.velo,
  dge.recep.velo,
  lig.fac,
  recep.fac
)
```

`addIntrDB` *Add interaction database to CytoSignal object*

Description

Add interaction database to CytoSignal object

Usage

```
addIntrDB(
  object,
  gene_to_uniprot,
  intr.db.diff_dep,
  intr.db.cont_dep,
  inter.index
)
```

Arguments

object	CytoSignal object
gene_to_uniprot	df, gene to uniprot symbols mapping
intr.db.diff_dep	intr.db for diffusable ligands + non-diffusable receptors
intr.db.cont_dep	intr.db for contact dependent ligands + receptors
inter.index	df, collection of intraction information

Value

a CytoSignal object

---

addVelo *Add velocity data to CytoSignal object*

Description

Add velocity data to CytoSignal object

Usage

```
addVelo(object, velo.s, velo.u)
```

Arguments

object	CytoSignal object
velo.s	matrix of spliced velo
velo.u	matrix of unspliced velo

changeUniprot *Subset gene expression matrix according to availability in the UNIPROT database.*

Description

Subset genes to those available in g_to_u, the gene name to UNIPROT ID databse provided with addIntrDB

Usage

```
changeUniprot(object, ...)

## S3 method for class 'matrix_like'
changeUniprot(object, gene_to_uniprot, verbose = TRUE, ...)

## S3 method for class 'CytoSignal'
changeUniprot(object, verbose = TRUE, ...)
```

Arguments

object	A CytoSignal object or a matrix.
...	Parameters passed to S3 methods.
gene_to_uniprot	Please use g_to_u.
verbose	Whether to show information of the progress.

Value

CytoSignal object with subset matrix updated in counts slot, or the subset matrix.

createCytoSignal *Create a CytoSignal object*

Description

Create a CytoSignal object

Usage

```
createCytoSignal(
  raw.data,
  cells.loc,
  clusters = NULL,
  name = NULL,
  parameters = NULL,
  log = NULL
)
```

Arguments

raw.data	raw data matrix
cells.loc	cells location matrix
clusters	cluster information
name	name of the dataset
parameters	parameters used
log	log of the processing

Value

a CytoSignal object

CytoSignal-class *The CytoSignal Class*

Description

The CytoSignal object is created from one spatial transcriptomic dataset. A CytoSignal object can be created with createCytoSignal.

Slots

counts Raw gene expression count matrix, sparse dgCMatrix, gene x cell.

cell.loc Matrix of cell locations, cell by axis.

clusters cluster assignments, named factor.

imputation List of imputation objects.

lrscore List of lrScore objects.

velo List of velo objects.

intr.valid List of interaction database. Added with addIntrDB.

cell.data data.frame of cell data.

parameters List of parameters for gaussian imputation

log Log of each main steps

version Version of package used to create object. Should not be modified by users.

db.diff *Interaction database derived from CellphoneDB V2*

Description

The referencing database that contains the list of curated interactions, the meta-information of each interaction, the ligand and receptor of each interaction, the protein components of both part of each interaction. The protein components are presented with UNIPROT ID, so we also provide g_to_u that maps UNIPROT ID to gene names.

Usage

```
db.diff

db.cont

inter.index

g_to_u
```

Format

db.diff and db.cont are lists of factor objects. db.diff is for diffusion dependent interactions and db.cont is for contact dependent interactions. Both lists have the following three entries of the same format:

- $ligands - A factor object where the values are IDs of interactions and names are UNIPROT IDs of the ligand protein components that match to the interaction.
- $receptors - A factor object of the same format as $ligands but for the receptor part of the interactions.
- $combined - The concatenation of the two factors above.

inter.index is a data.frame object with the meta-information of each interaction.

g_to_u is a data.frame object that map gene names to UNIPROT ID of their protein product.

An object of class list of length 3.

An object of class data.frame with 1396 rows and 11 columns.

An object of class data.frame with 977 rows and 3 columns.

Source

CellphoneDB V2

findNN *Identify nearest neighbors for each location using different strategies for different types of interactions*

Description

This is a wrapper function of findNNGauEB, findNNDT and findNNRaw. We use a Gaussian Epsilon ball to identify the nearest neighbors that can contribute to the diffusible ligands. And then uses a Delaunay triangulation to identify the nearest neighbors that can contribute to the contact-dependent ligands. The identified nearest neighbors will then be used for imputing the L or R value for each location.

Usage

```
findNN(
  object,
  eps = NULL,
  sigma = NULL,
  self.weight = "auto",
  weight = 2,
  max.r = NULL
)
```

Arguments

object	A CytoSignal object.
eps	Gaussian Epsilon Ball method parameter. A numeric scalar. The radius of the Gaussian Epsilon ball. Default NULL uses parameter inferred with inferEpsParams.
sigma	Gaussian Epsilon Ball method parameter. The σ of the Gaussian kernel. Default NULL uses parameter inferred with inferEpsParams.
self.weight	Gaussian Epsilon Ball method parameter. Weight of the index cell. Use a number between 0-1 or a string "auto" or "sum_1". Default "auto".
weight	Delaunay Triangulation method parameter. A numeric scalar for the sum of the weights of the edges of the Delaunay Triangulation. Default 2.
max.r	Delaunay Triangulation method parameter. A numeric scalar for the maximum radius of the edges. Default NULL uses parameter inferred with inferEpsParams.

Value

A CytoSignal object updated. object@imputation slot will be updated with three new entries: object@imputation$GauEps, object@imputation$DT and object@imputation$Raw.

Examples

```
## Not run:
object <- findNN(object)

## End(Not run)
```

`findNNDT` *Find the direct connected neighbor of each cell, using Delaunay triangulation*

Description

Find the direct connected neighbor of each cell, using Delaunay triangulation

Usage

```
findNNDT(object, ...)

## S3 method for class 'matrix'
findNNDT(object, weight.sum = 2, max.r = NULL, ...)

## S3 method for class 'CytoSignal'
findNNDT(object, weight = 2, max.r = NULL, ...)
```

Arguments

`object`	A CytoSignal object or a matrix of cell location.
`...`	Arguments passed to other S3 methods
`weight.sum`	The sum of the weights
`max.r`	The maximum radius of the edges, by default is the r.diffuse.scale,

Value

For CytoSignal object, the original object with result updated. For matrix object, a list of neighbors.

findNNGauEB *Find the neighbors of each cell in the Epsilon Ball*

Description

Find the neighbors of each cell in the Epsilon Ball

Usage

```
findNNGauEB(object, eps = NULL, sigma = NULL, self.weight = "auto", ...)

## S3 method for class 'matrix'
findNNGauEB(object, eps, sigma = 0.15, self.weight = "auto", ...)

## S3 method for class 'CytoSignal'
findNNGauEB(
  object,
  eps = NULL,
  sigma = NULL,
  self.weight = "auto",
  tag = NULL,
  ...
)
```

Arguments

object	A CytoSignal object or a matrix object for the cell spatial location.
eps	The radius of the epsilon ball. For CytoSignal object, by default use parameter inferred with inferEpsParams.
sigma	The sigma of the Gaussian kernel. Default 0.15 for matrix method. For CytoSignal object, by default use parameter inferred with inferEpsParams.
self.weight	weight of the index cell. Use a number between 0-1, or choose from "auto" or "sum_1".
...	Arguments passed to other S3 methods.
tag	Name prefix of the analysis.

Value

For CytoSignal object, the orignal object with result updated. For matrix object, a list of neighbors and their distances.

findNNRaw *Create a ImpData object using raw data without imputation*

Description

Create a ImpData object using raw data without imputation

Usage

```
findNNRaw(object)
```

Arguments

object	A Cytosignal object

Value

A Cytosignal object

format.CytosignalIntrDEG *Format a CytosignalIntrDEG object to string*

Description

Format a CytosignalIntrDEG object to string

Usage

```
## S3 method for class 'CytosignalIntrDEG'
format(x, ...)
```

Arguments

x	A CytosignalIntrDEG object.
...	Passed to other methods

Value

A string representation of the object

graphNicheLR *Compute the LR score for specific ligand-receptor imputation obj pairs*

Description

Compute the LR score for specific ligand-receptor imputation obj pairs

Usage

```
graphNicheLR(object, ...)
```

Arguments

object	A Cytosignal object
lig.slot	The ligand slot to use
recep.slot	The receptor slot to use
intr.db.name	The intr database name to use
nn.use	The neighbor index as niche

Value

A Cytosignal object

graphNicheLR.CytoSignal *Sub function for graphNicheLR, input a CytoSignal object*

Description

Sub function for graphNicheLR, input a CytoSignal object

Sub function for graphNicheLR, input a CytoSignal object

Usage

```
## S3 method for class 'CytoSignal'
graphNicheLR(object, lig.slot, recep.slot, intr.db.name, nn.use = NULL)

## S3 method for class 'CytoSignal'
graphNicheLR(object, lig.slot, recep.slot, intr.db.name, nn.use = NULL)
```

Arguments

object	A Cytosignal object
lig.slot	The ligand slot to use
recep.slot	The receptor slot to use
intr.db.name	The intr database name to use
nn.use	slot that the neighbor index should be taken from, by default is the same as the recep.slot. For example, if score.obj = GauEps-DT, then nn.use = "DT". nn.use could also be a user-defind factor.

Value

A Cytosignal object

A Cytosignal object

---

graphNicheLR.dgCMatrix

Sub function for graphNicheLR, input a sparse matrix

Description

Sub function for graphNicheLR, input a sparse matrix

Usage

```
## S3 method for class 'dgCMatrix'
graphNicheLR(dge.lig, dge.recep, nb.id.fac, lig.fac, recep.fac)
```

Arguments

dge.lig	A sparse matrix for ligand
dge.recep	A sparse matrix for receptor
nb.id.fac	A factor of neighbor indices
lig.fac	A factor of ligand indices
recep.fac	A factor of receptor indices

Value

A sparse matrix

heatmap_GO *Show significant genes across top GO term hits with coefficients from regression analysis of an interaction*

Description

Create a heatmap for an interaction on GO terms by significant genes, colored by the coefficients of the genes in the regression model returned by `inferIntrDEG`. The GO term enrichment can be done with any tools available. The input data.frame `GO` must contain fields for 1. term description, character, pointed to by `description.col`, 2. p-value, numeric, pointed to by `pval.col`, 3. gene hit string, character, pointed to by `gene.col`. The gene hit string for each term must be form in a way that function `gene.split.fun` can split it into a character vector of gene names. For example, if a gene hit string is "gene1, gene2, gene3", then `gene.split.fun` should be `function(x) unlist(strsplit(x, ", "))`, so that a split result `c("gene1", "gene2", "gene3")` can be correctly obtained.

Usage

```
heatmap_GO(
  intrDEG,
  GO,
  intr,
  description.col = "description",
  pval.col = "pval",
  gene.col = "genes",
  gene.split.fun = function(x) unlist(strsplit(x, ",")),
  term.topN = 20,
  gene.topN = 20,
  binary_sign = FALSE,
  text.size = 10
)
```

Arguments

`intrDEG`	A `CytosignalIntrDEG` object, output from `inferIntrDEG`.
`GO`	A data.frame object for GO enrichment analysis result.
`intr`	A single interaction ID (starts with "CPI") or its numeric index within the range of `intrDEG`.
`description.col, pval.col, gene.col`	The column names of the data.frame `GO` that contains the term description, p-value, and gene hit string. Default `"description"`, `"pval"`, `"genes"`.
`term.topN`	Use this number of top GO terms, ranked by p-values. Default `20`.
`gene.topN`	Use this number of top genes, ranked by absolute value of coefficients. Default `20`.
`binary_sign`	Whether to convert coefficient value to binary sign value. Default `FALSE`.

color_num — Number of colors in the heatmap. Can only use 2 or 3, Default 2 use white-red color palette. 3 use scaled blue-white-red color palette.

---

hex_bin — *GGPLOT2 FUNCTIONALITY FOR MAPPING TO HEXAGON SIZE AND COLOUR AESTHETICS by Robin Edwards, 2013 (geotheory.co.uk, @geotheory) This has been adapted from the ggplot bin_hex.R script that underpins geom_hex, etc (see https://github.com/hadley/densityvis/blob/master/R/bin-hex.r).*

Description

These functions implement aesthetic mapping to hexagon size (area), in addition to the existing colour-mapping functionality. The key change is the addition of a new fourth variable (var4) to hex_bin(), which complements the inbuilt hexagon binning functionality. The 'frequency.to.area' argument enables the default mappings of binned data to colour and var4 to size to be interchanged. The hmin/hmax arguments 0,1 set area mapping constraints (hmax can exceed 1). xlim/xlat enable hexagon tesselation to be constrained independently of data range. There may be some bugs in the implementation. A legend for hexagon size has not been implemented. Bin data into hexagons (2d).

Usage

```
hex_bin(
  x,
  y,
  weight = NULL,
  var4 = NULL,
  width = NULL,
  height = NULL,
  xbins = 20,
  ybins = 20,
  var4.to.color = FALSE,
  na.rm = FALSE,
  hmin = 0,
  hmax = 1,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

x	a numeric vector of x positions
y	a numeric vector of y positions
weight	NULL or a numeric vector providing weights for each observation, replace counts, mapped to color

var4	NULL or a numeric vector providing weights for each observation, averaged within each bin, mapped to hex bin size
width	width of each hexagon, if NULL computed from ybins
height	height of each hexagon, if NULL computed from ybins
xbins	number of horizontal bins, if width unspecified
ybins	number of vertical bins, if height unspecified
na.rm	If TRUE missing values will be silently removed, otherwise they will be removed with a warning.

Value

A data frame with columns x, y and freq, and attributes width and height.

See Also

hex_pos for algorithm that finds hexagon center closest to each point and hex_coord that generates coordinates of each hexagon.

Examples

```
plot(hex_bin(runif(1e4), runif(1e4)))
plot(hex_bin(rnorm(1e4), rnorm(1e4)))

data(baseball, package = "plyr")
bin <- hex_bin(baseball$g, baseball$ab)
```

ImpData-class *The ImpData Class*

Description

The ImpData object is created from one ST dataset. User could choose a preferred imputation method and the class stores the imputed data, the imputed normalized data, the intr database, the intr database.

Details

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

Slots

method imputation method

imp.data imputed data

imp.norm.data imputed normalized data

intr.data imputed normalized data subsetted by intr database

`intr.data.null` permuted imputed normalized data subsetted by intr database

`nn.id` nearest neighbor id

`nn.dist` nearest neighbor distance

`log` Log of each main steps

`imputeLR` *Impute the L or R value from the nearest neighbors of each location*

Description

After running `findNN`, we can impute the L or R value from the nearest neighbors of each location basing on the types of nearest neighbors.

Usage

```
imputeLR(object, weights = c("none", "mean", "counts", "dist"))
```

Arguments

`object`	A CytoSignal object, with `findNN` already run.
`weights`	A character scalar. The method to calculate the weights of the Delauany Triangulation. Choose from `"none"`, `"mean"`, `"counts"` and `"dist"`. Default `"none"`.

Value

A CytoSignal object updated. Entries in `object@imputation` slot will be updated with the imputation values.

Examples

```
## Not run:
object <- findNN(object)
object <- imputeLR(object)

## End(Not run)
```

`imputeNiche` *Impute the data*

Description

Impute the data

Usage

```
imputeNiche(object, weights = c("mean", "counts", "dist", "none"), ...)

## S3 method for class 'dgCMatrix'
imputeNiche(
  object,
  nb.id.fac,
  nb.dist.fac,
  weights = c("mean", "counts", "dist", "none"),
  ...
)

## S3 method for class 'CytoSignal'
imputeNiche(
  object,
  nn.type = NULL,
  weights = c("mean", "counts", "dist", "none"),
  ...
)
```

Arguments

`object`	A Cytosignal object or a dgCMatrix object of raw gene expression matrix
`weights`	The weight of the Delaunay triangulation. Choose from `"mean"`, `"counts"`, `"dist"` or `"none"`.
`...`	Arguments passed to other S3 methods
`nb.id.fac`	A factor of neighbors
`nb.dist.fac`	A factor of weights
`nn.type`	The type of neighbors. The `tag` used when finding nearest neighbors in the upstream step. If not customized, use `"GauEps"` for NN found with `findNNGauEB`, or use `"DT"` for NN found with `findNNDT`. Default use the most lastly computed NN.

Value

For CytoSignal object, the original object with imputation result updated. For dgCMatrix object, a sparse N x N graph presented as another dgCMatrix object.

`imputeNicheVelo` *Impute the velocity mtx using the specified method*

Description

Impute the velocity mtx using the specified method

Usage

```
imputeNicheVelo(object, ...)
```

Arguments

`object`	A Cytosignal object
`...`	Other parameters
`method`	The method to use for imputation

Value

A Cytosignal object

`imputeNicheVelo.CytoSignal` *Sub function for imputeNicheVelo, input a Cytosignal object*

Description

Sub function for imputeNicheVelo, input a Cytosignal object

Usage

```
## S3 method for class 'CytoSignal'
imputeNicheVelo(object, nn.type = NULL)
```

Arguments

`object`	A Cytosignal object
`nn.type`	The type of neighbors

Value

A Cytosignal object

imputeVeloLR — *Impute time derivative of L or R from the nearest neighbors of each location*

Description

After running `findNN`, we can impute the temporal L or R change from the nearest neighbors of each location basing on the types of nearest neighbors.

Usage

```
imputeVeloLR(object)
```

Arguments

`object`	A CytoSignal object, with `findNN` already run and velocity information added with `addVelo`.

Value

A CytoSignal object updated. Entries in `object@imputation` slot will be updated with the imputation values.

Examples

```
## Not run:
object <- findNN(object)
object <- imputeVeloLR(object)

## End(Not run)
```

inferCorrScore — *Infer the correspondence between LR-scores and Significance*

Description

Min-max the LRscores, substract the significant ones, sum and average.

Usage

```
inferCorrScore(object, correctBy = c("cell", "intr"), slot.use = NULL)
```

Arguments

`object`	A Cytosignal object
`correctBy`	Spatial FDR correction method. Choose from `"cell"` or `"intr"`.
`slot.use`	Which LR score to use. Use the name specified with `tag` when running `inferLRScore`.

Value

A Cytosignal object

`inferEpsParams` *Infer the parameters of the Gaussian kernel*

Description

Infer the parameters of the Gaussian kernel

Usage

```
inferEpsParams(object, scale.factor = NULL, r.eps.real = 200, thresh = 0.001)
```

Arguments

`object`	A Cytosignal object
`r.eps.real`	The radius of the epsilon ball in tech resolution in um, default 200 um
`thresh`	The total signal out of the epsilon ball
`scale.facor`	1 spatial coord unit equals to how many μm

Value

A Cytosignal object

`inferIntrDEG` *Infer significant genes for each interaction*

Description

This function performs wilcoxon one-sided test for each cell type, between cells enriched with an interaction against other cells. The top significant genes are selected for each interaction. The expression profile of the selected genes, together with the cell type variable, are then feeded into a regression model for further refinement. This analysis infers the significant genes for each interaction, and refines the LRScore using the selected genes.

Usage

```
inferIntrDEG(
  object,
  intr = NULL,
  slot.use = NULL,
  signif.use = NULL,
  num.per.choose = 50,
  alpha.test = seq(0.5, 1, 0.1),
  seed = 1,
  minCell = 50,
  verbose = TRUE
)
```

Arguments

object	A Cytosignal object with inferIntrScore already run.
intr	Specify interactions to be used. A vector of either the unique ID of interactions or the numeric rank indices. Available IDs can be shown with showIntr(object). Availability of an interaction depends on the LRscore slot to be used as well as the significance metric to be used.
slot.use	The LRscore type to be used. See vignette for explanation.
signif.use	The significance metric to be used. See vignette for explanation.
num.per.choose	The maximum number of genes to be chosen for each wilcoxon test. A test happens for each interaction and each cell type. Default 50.
alpha.test	A sequence of alpha values to be tested for the elastic net regression model. The best alpha value will be chosen based on the smallest loss. Larger alpha values result in less final selection. Default seq(0.5, 1, 0.1).
seed	Random seed for controlling the cross validation. Default 1.
minCell	Minimum number of cells where a gene is expressed to be considered. Default 50.
verbose	Whether to print progress messages. Default TRUE.

Value

list object, each element is the result for each interaction. Each interaction result is a list object containing the following entries:

- glmnet_model - A "cv.glmnet" object, the regression model.
- score_refine - A 1-column matrix of refined LR scores of this interaction in each cell.
- coef - A matrix of coefficients of the regression model.
- sign_clusters - A character vector of significant clusters in selected by the model.
- sign_genes - A character vector of significant genes selected by the model.
- alpha - The alpha value used for the final chosen model.
- slot.use - The LRScore type used for this analysis.
- signif.use - The significance metric used for this analysis.

`inferIntrScore` *Calculate LRScore from the imputed L and R values*

Description

After running `imputeLR`, we can calculate the LR score for each location. The LR score is calculated as the product of the imputed L and R values. With the LR score inferred, we subsequently perform permutation tests to construct the null distribution for testing the significance of the interactions.

Usage

```
inferIntrScore(
  object,
  recep.smooth = FALSE,
  intr.type = c("diff", "cont"),
  perm.size = 1e+05,
  numCores = 1
)
```

Arguments

`object`	A CytoSignal object, with `imputeLR` already run.
`recep.smooth`	A logical scalar. Whether to use the smoothed R values which is imputed with DT method. Default `FALSE`.
`intr.type`	A character vector. The type of interactions to calculate the LR score. Choose from one or both of `"diff"` and `"cont"` for diffusion-dependent and contact-dependent interactions, respectively. Default uses both.
`perm.size`	A numeric scalar. The number of permutations to perform. Default `1e5` times.
`numCores`	SPARK::sparkx parameter. The number of cores to use. Default `1`.

Value

A CytoSignal object updated. Entries in `object@lrscore` slot will be updated with the LR scores and the significance inferrence. When `recep.smooth` is by default `FALSE`, the `object@lrscore` slot will be updated with `object@lrscore$`GauEps-Raw`` and `object@lrscore$`DT-Raw``. When `recep.smooth` is `TRUE`, the `object@lrscore` slot will be updated with `object@lrscore$`GauEps-DT`` and `object@lrscore$`DT-DT``.

Examples

```
## Not run:
object <- findNN(object)
object <- imputeLR(object)
object <- inferIntrScore(object)

## End(Not run)
```

`inferIntrVelo` *Calculate the interaction velocity from the imputed time derivative of L or R values*

Description

After running `imputeVeloLR`, we can calculate the interaction velocity for each location. Please refer to the manualscript for detail of the calculation.

Usage

```
inferIntrVelo(object, recep.smooth = FALSE, norm.method = "scanpy")
```

Arguments

`object`	A CytoSignal object, with `imputeVeloLR` already run.
`recep.smooth`	A logical scalar. Whether to use the smoothed R values which is imputed with DT method. Default `FALSE`.
`norm.method`	The normalization method to apply to the velocity data, need to be consistent with the normalization method used when generating the input RNA velocity. Please consult the tool used for it, e.g. scVelo, VeloVAE. Default is `"scanpy"`. Can choose from `"scanpy"`, `"cpm"`, `"default"` or `"none"`.

Value

A CytoSignal object updated. Entries in `object@lrvelo` slot will be updated with the velocity scores. When `recep.smooth` is by default `FALSE`, the `object@lrvelo` slot will be updated with `object@lrvelo$` GauEps-Raw` and `object@lrvelo$` DT-Raw`. When `recep.smooth` is `TRUE`, the `object@lrvelo` slot will be updated with `object@lrvelo$` GauEps-DT` and `object@lrvelo$` DT-DT`.

Examples

```
## Not run:
object <- addVelo(object, velo.s, velo.u)
object <- findNN(object)
object <- imputeVeloLR(object)
object <- inferIntrVelo(object)

## End(Not run)
```

inferNullScoreLR *Permute LR score for specific ligand-receptor imputation obj pairs*

Description

This function is a follow-up function of inferScoreLR. It computes the NULL LR-scores using the NULL imputation results and stores the results in the LR score object. The null distribution of the LR scores can be used to test the significance of the LR scores.

Usage

```
inferNullScoreLR(object, slot.use = NULL)
```

Arguments

object	A Cytosignal object
slot.use	Which LR score to use. Use the name specified with tag when running inferLRScore.

Value

A Cytosignal object

inferScoreLR *Compute the LR score for specific ligand-receptor imputation pairs*

Description

Compute the LR score for specific ligand-receptor imputation pairs

Usage

```
inferScoreLR(
  object,
  lig.slot,
  recep.slot,
  intr.db.name = c("diff_dep", "cont_dep"),
  tag = paste0(lig.slot, "-", recep.slot)
)
```

Arguments

object	A Cytosignal object
lig.slot	The ligand slot to use
recep.slot	The receptor slot to use
intr.db.name	The intr database name to use
tag	Name of the result to be stored in object.

Value

A Cytosignal object

`inferSignif` *Infer significance of LR scores*

Description

Infer significance of LR scores

Usage

```
inferSignif(
  object,
  fdr.method = c("spatialFDR", "fdr"),
  p.value = 0.05,
  reads.thresh = 100,
  sig.thresh = 100,
  slot.use = NULL,
  nn.use = NULL
)
```

Arguments

`object`	A Cytosignal object
`fdr.method`	The false discovery rate method to use. Choose from `"spatialFDR"` and `"fdr"`. Default `"spatialFDR"`.
`p.value`	A numeric scalar. The p-value threshold to use for filtering significant interactions. Default `0.05`.
`reads.thresh`	A numeric scalar. The minimum number of reads for a ligand-receptor interaction to be considered. Default `100`.
`sig.thresh`	A numeric scalar. The minimum number of of beads for a ligand-receptor interaction to be considered. Default `100`.
`slot.use`	Which LR score to use. Use the name specified with `tag` when running `inferIntrScore`. Default `NULL` apply specified significance and filtering criteria to all available LRscore slots.
`nn.use`	Which imputation to use. Default the imputation used for deriving the LRScore specified with `slot.use`. Use the name specified with `tag` when running `findNNGauEB`; use `"DT"` for imputation produced with `findNNDT`; or use `"Raw"` for imputation produced with `findNNRaw`.

Value

A Cytosignal object

inferSpatialCorr *Infer the correspondence between LR-scores and Significance*

Description

Use DT neighbors to impute the p-value and LR-score for each cell, then compute the pearson correlation between the imputed LR-score and p-value.

Usage

```
inferSpatialCorr(object, correctBy = c("cell", "intr"), slot.use = NULL)
```

Arguments

object	A Cytosignal object
correctBy	Spatial FDR correction method. Choose from "cell" or "intr".
slot.use	Which LR score to use. Use the name specified with tag when running inferLRScore.

Value

A Cytosignal object

inferVeloLR *Sub function for inferVeloLR, input a CytoSignal object*

Description

Sub function for inferVeloLR, input a CytoSignal object

Usage

```
inferVeloLR(
  object,
  lig.slot,
  recep.slot,
  intr.db.name,
  norm.method = "scanpy",
  tag = NULL
)
```

Arguments

object	A Cytosignal object
lig.slot	The ligand slot to use
recep.slot	The receptor slot to use
intr.db.name	The intr database name to use
norm.method	The normalization method to apply to the counts, need to be consistent with the normalization method used for the RNA velocity method. Default is "scanpy".
nn.use	slot that the neighbor index should be taken from, by default is the same as the recep.slot. For example, if velo.obj = GauEps-DT, then nn.use = "DT". nn.use could also be a user-defind factor.

Value

A Cytosignal object

---

lrScores-class *The lrScores Class*

Description

The lrScores object is created from one ST dataset. User could choose two imputation methods to calculate the ligand-receptor scores. The class stores the ligand, receptor, and interaction database, the ligand-receptor scores, the ligand-receptor scores for permuted data, the ligand-receptor scores for permuted data, and the log of each main steps.

Details

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

Slots

`lig.slot` ligand database

`recep.slot` receptor database

`intr.slot` interaction database

`intr.list` list of interaction database

`score` ligand-receptor scores

`score.null` permuted ligand-receptor scores

`res.list` list of results

`log` Log of each main steps

lrVelo-class *The lrvelo Class*

Description

The lrvelo object is created from one ST dataset. User could choose two imputation methods to calculate the ligand-receptor scores. The class stores the index of ligand, receptor, and interaction database, inferred lrvelo for each interaction, and the log of each main steps.

Details

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

Slots

`lig.slot` ligand database

`recep.slot` receptor database

`intr.slot` interaction database

`intr.list` list of interaction database

`velo.s` A matrix of spliced velo for each gene

`velo.u` A matrix of unspliced velo for each gene

`velo.intr` A sparse matrix of velo for each intr

`nn.id` A factor of nearest neighbor id. Re-do findNN since the order of cells may change!

`nn.dist` A factor of nearest neighbor distance. Re-do findNN since the order of cells may change!

`log` Log of each main steps

permuteLR *Permute Imputation Results of specific imputation method*

Description

This function permutes the imputation methods from which a given LR score is calculated. Note that all rounds of permutation will use the same shuffle and sample index.

Usage

```
permuteLR(object, slot.use = NULL, norm.method = "default", perm.size = 1e+05)
```

Arguments

`object`	A CytoSignal object with `inferLRScore` run beforehand.
`slot.use`	Which LR score to use. Use the name specified with `tag` when running `inferLRScore`.
`perm.size`	Size of the permutation test. Defult `100000`.
`norm.me`	Normalization method. See `normCounts` for detail.

Value

A CytoSignal object

`plotEdge` *Plotting edge for a given interaction from a CytoSignal object*

Description

Plotting edge for a given interaction from a CytoSignal object

Usage

```
plotEdge(
  object,
  intr,
  type = c("receiver", "sender"),
  slot.use = NULL,
  signif.use = NULL,
  colors.list = NULL,
  return.plot = TRUE,
  plot_dir = "csEdgePlot/",
  filename = NULL,
  plot.fmt = c("png", "pdf", "svg"),
  title = NULL,
  edge.size = 500,
  use.shape = 16,
  line.width = 0.01,
  use.phi = 30,
  use.theta = -17,
  z.scaler = 0.03,
  box = TRUE,
  z.pt.interval = 1,
  pt.size = 0.1,
  pt.sig.size = NULL,
  pt.stroke = 0.2,
  width = 5,
  height = 5,
  set.res = 300,
  verbose = TRUE
)
```

Arguments

`object`	`CytoSignal` object.
`intr`	Interaction to plot. See available options with `showIntr`.
`type`	Type of plot, either `"sender"` or `"receiver"`. Default `"sender"`.

slot.use	Slot to use for plotting
signif.use	Significance level to use for plotting
colors.list	List of colors to use for plotting
return.plot	Whether to return "plist" object for figure organization. TRUE returns "plist" which can be shown on current display device with plot(). FALSE saves the figure to disk. Default TRUE.
plot_dir	Path where the figure will be saved when return.plot = FALSE. Default "csEdgePlot/" (under current working directory).
filename	Filename of the figure inside plot_dir. Please match extension name with plot.fmt and do not include path. Default NULL and the exact filename will be determined by interaction information and plot.fmt.
plot.fmt	Format of plot, either "png" or "pdf".
title	Title of plot
use.shape	Shape of points
line.width	Width of lines
use.phi	Angle of 3D plot
use.theta	Angle of 3D plot
z.scaler	Scaling factor for z-axis
z.pt.interval	Interval of points on z-axis
pt.stroke	Width of points
set.res	Resolution of plot
use.cex	Size of points
u_width	Width of plot
u_hgt	Height of plot

Value

plist object when return.plot = TRUE, no in memory object is returned when return.plot = FALSE but the figure will be saved on disk.

---

plotRefinedScore *Plot the refined score of each interaction after regression model refinement*

Description

Plot the refined score of each interaction after regression model refinement

Usage

```
plotRefinedScore(
  object,
  intrDEGRes,
  intr = NULL,
  pt.size = 0.5,
  pt.stroke = 0.1
)
```

Arguments

object	A Cytosignal object with inferIntrScore already run.
intrDEGRes	The direct output object of inferIntrDEG.
intr	A vector of unique interaction IDs that are available in intrDEGRes, or numerical index within its range. Default NULL use all the results.
pt.size	Size of the points in the plot. Default 0.5.
pt.stroke	Stroke size of the points in the plot. Default 0.1.

Value

List of ggplot objects, each shows the refined LR score of each selected interaction.

---

plotSigCluster *Plot edges by each cluster as sender or receiver cells*

Description

By default ranked by the number of significant cells in each cluster

Usage

```
plotSigCluster(
  object,
  plot_dir,
  cluster.list = NULL,
  intr.num = 10,
  type = c("sender", "receiver"),
  slot.use = NULL,
  signif.use = NULL,
  colors.list = NULL,
  plot.fmt = "png",
  edge.size = 2000,
  all.in.one = T,
  plot.all.sig = F,
  use.cex = 0.1,
  use.shape = 16,
```

```
  line.width = 0.02,
  use.phi = 30,
  use.theta = -17,
  z.scaler = 0.03,
  z.pt.interval = 1,
  pt.stroke = 0.2,
  u_width = 6,
  u_hgt = 6,
  set.res = 400,
  return.plot = F
)
```

Arguments

object	A CytoSignal object
plot_dir	Directory to save plots
cluster.list	A list of clusters to plot
intr.num	Number of interactions to plot
type	Either "sender" or "receiver"
slot.use	Slot to use for plotting
signif.use	Significance threshold to use for plotting
colors.list	A list of colors to use for each cluster
plot.fmt	Plot format
edge.size	Number of edges to plot
all.in.one	Plot all clusters in one plot
plot.all.sig	Plot all significant edges
use.shape	Shape of points
line.width	Width of lines
use.phi	Angle of view
use.theta	Angle of view
z.scaler	Scale of z-axis
z.pt.interval	Interval of z-axis
pt.stroke	Stroke of points
u_width	Width of plot
u_hgt	Height of plot
set.res	Resolution of plot
return.plot	Return plot object
pt.size	Size of points
...	Other arguments

Value

A list of plots

`plotSignif` *Plot significant interactions ranked by the user-specified metric*

Description

Plot significant interactions ranked by the user-specified metric

Usage

```
plotSignif(
  object,
  num.plot = NULL,
  res_dir,
  plot.details = T,
  slot.use = NULL,
  signif.use = NULL,
  plot.clusters = T,
  plot.velo = F,
  colors.list = NULL,
  pt.size = 0.1,
  pt.stroke = 0.2,
  u_width = 6,
  u_hgt = 5,
  set.res = 200,
  return.plot = F
)
```

Arguments

`object`	A cytosignal object
`num.plot`	Number of interactions to plot
`res_dir`	Directory to save the plots
`plot.details`	Whether to plot NULL imputed values and scores
`slot.use`	The LRscore slot to use for plotting
`signif.use`	The metric used to rank the interactions, by default "result.hq.pear"
`plot.clusters`	Whether to plot the clusters
`plot.velo`	Whether to plot the velocity
`colors.list`	A list of colors to use for plotting
`pt.size`	Size of the points
`pt.stroke`	Stroke of the points
`u_width`	Width of the plot
`u_hgt`	Height of the plot
`set.res`	Resolution of the plot
`return.plot`	Whether to return the plot

Value

A plot if return.plot is TRUE. Otherwise, plots are saved to the specified directory.

`plotSignif2` *Plot significant interactions ranked by the user-specified metric*

Description

Plot significant interactions ranked by the user-specified metric

Usage

```
plotSignif2(
  object,
  intr,
  edge = FALSE,
  velo = FALSE,
  slot.use = NULL,
  signif.use = NULL,
  colors.list = NULL,
  pt.size = 0.1,
  pt.stroke = 0.2,
  return.plot = FALSE,
  plot_dir = "csSignifPlot/",
  plot.fmt = c("png", "pdf", "svg"),
  raster = NULL,
  resolution = 300,
  verbose = FALSE
)
```

Arguments

`object`	A cytosignal object
`intr`	Specify interactions to be plotted. A vector of either the unique ID of interactions or the numeric indices. Available IDs can be shown with `showIntr(object)`. Availability of an interaction depends on the LRscore slot to be used as well as the significance metric to be used.
`edge`, `velo`	Logical, whether to plot edge or velocity, respectively.
`slot.use`	The LRscore slot to use for plotting
`signif.use`	The metric used to rank the interactions, by default "result.hq.pear"
`colors.list`	A list of colors to use for plotting
`pt.size`	Size of the points
`pt.stroke`	Stroke of the points
`return.plot`	Whether to return the plot

plot_dir	Directory to save the plots
plot.fmt	Format of output file. "png", "pdf", or "svg".
resolution	Resolution of the output figure.
verbose	Logical, whether to show progress.

Value

A plot if return.plot is TRUE. Otherwise, plots are saved to the specified directory.

---

plotVelo *Plot 3D LR-velo ranked by the user-specified metric*

Description

Plot 3D LR-velo ranked by the user-specified metric

Usage

```
plotVelo(
  object,
  intr,
  return.plot = TRUE,
  plot_dir = "csVeloPlot/",
  filename = NULL,
  plot.fmt = c("png", "pdf"),
  slot.use = NULL,
  signif.use = NULL,
  use.clusters = NULL,
  colors.list = NULL,
  z.scaler = 0.03,
  title = NULL,
  pt.size = 0.1,
  use.shape = 16,
  use_xbins = 15,
  use_ybins = 15,
  arrow.line.width = 0.6,
  arrow.width = 0.06,
  pt.stroke = 0.2,
  use.phi = 30,
  use.theta = -17,
  box = TRUE,
  axis.arrow.len = 1,
  width = 6,
  height = 6,
  set.res = 300,
  verbose = TRUE
)
```

Arguments

object	A cytosignal object
return.plot	Whether to return the plot
slot.use	The LRscore slot to use for plotting
signif.use	The metric used to rank the interactions, by default "result.hq.pear"
use.clusters	Plot only selected clusters. Default NULL for all clusters.
colors.list	A list of colors to use for plotting
z.scaler	Scaling factor for the z-axis
pt.size	Point size. Default 0.1
use.shape	Point shape. Default 16
use_xbins	Number of bins for the x-axis. Default 15
use_ybins	Number of bins for the y-axis. Default 15
arrow.line.width	Width of the arrow line. Default 0.6
arrow.width	Width of the arrow. Default 0.06
pt.stroke	Stroke of the points
use.phi	Set view angel: phi the colatitude. Default 30
use.theta	Set view angel: theta gives the azimuthal direction. Default -17
box	Whether to show a box panel for the 3D region. Default TRUE
axis.arrow.len	When box = FALSE, set the length of the axis arrows.
width	Width of the plot. Default 6
height	Height of the plot. Default 6
set.res	Resolution of the plot. Default 300.
num.plot	Number of interactions to plot
res_dir	Directory to save the plots
plot.velo	Whether to plot the velocity

Value

A plot if return.plot is TRUE. Otherwise, plots are saved to the specified directory.

print.CytosignalIntrDEG *Print the CytosignalIntrDEG object representation to screen*

Description

Print the CytosignalIntrDEG object representation to screen

Usage

```
## S3 method for class 'CytosignalIntrDEG'
print(x, ...)
```

Arguments

x	A CytosignalIntrDEG object.
...	Passed to other methods

purgeBeforeSave *Remove imputed data and normalized imputed data from CytoSignal object to save disk space*

Description

Remove imputed data and normalized imputed data from CytoSignal object to save disk space

Usage

```
purgeBeforeSave(object, purge.raw = TRUE, purge.null = FALSE)
```

Arguments

object	CytoSignal object

Value

a CytoSignal object

rankIntrSpatialVar *Rank the inferred high-quality interactions by their spatial variability*

Description

This function utilizes SPARK package to calculate the spatial variability of the high-quality interactions, using their LR scores. Please refer to Jiaqiang Zhu, et al., 2021, Genome Biology for more details of the method.

Usage

```
rankIntrSpatialVar(object, slot.use = NULL, numCores = 1, verbose = FALSE)
```

Arguments

object	A CytoSignal object, with inferSignif already run.
slot.use	Which LR score to use. Use the name specified with tag when running inferLRScore.
numCores	SPARK::sparkx parameter, an integer specifying the number of threads.
verbose	SPARK::sparkx parameter, a logical value indicating whether to print details for debug purpose

Value

The input CytoSignal object with the spatially variable high-quality interaction list updated at `object@lrscore[[slot.use]]@res.list$result.spx`

Examples

```
## Not run:
object <- findNN(object)
object <- imputeLR(object)
object <- inferScoreLR(object, lig.slot = "GauEps", recep.slot = "Raw",
                       intr.db.name = "diff_dep")
object <- permuteLR(object)
object <- inferNullScoreLR(object)
object <- inferSignif(object)
object <- rankIntrSpatialVar(object)

## End(Not run)
```

`removeLowQuality` *Remove low quality cells and genes from raw counts*

Description

Remove low quality cells and genes from raw counts

Usage

```
removeLowQuality(object, counts.thresh = 300, gene.thresh = 50)
```

Arguments

`object`	CytoSignal object
`counts.thresh`	threshold for cell counts
`gene.thresh`	threshold for gene counts

Value

a CytoSignal object

`runPears.std` *Identify spatially significant interactions using std-corrected pearson correlation Normal Moran's I test is not applicable here since the total number of the cell is too large, causing unnacceptable computation cost. Here we use a modified version of Moran's I test, which is to take only the top KNNs to compute the Moran's I test.*

Description

Identify spatially significant interactions using std-corrected pearson correlation Normal Moran's I test is not applicable here since the total number of the cell is too large, causing unnacceptable computation cost. Here we use a modified version of Moran's I test, which is to take only the top KNNs to compute the Moran's I test.

Usage

```
runPears.std(object, k = 10, weight = 2, score.slot = NULL)
```

Arguments

`object`	A Cytosignal object
`k`	The number of nearest neighbors to use
`weight`	The weight of the nearest neighbors
`score.slot`	Which LR score to use. Use the name specified with `tag` when running `inferLRScore`.

Value

A Cytosignal object

`show,CytoSignal-method` *show method for CytoSignal*

Description

show method for CytoSignal

Usage

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

Arguments

`object`	CytoSignal object

`show,ImpData-method` *show method for cytosignal obj*

Description

show method for cytosignal obj

Usage

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

`showImp` *show method for ImpData*

Description

show method for ImpData

Usage

```
showImp(object, slot.use = NULL)
```

Arguments

`object`	CytoSignal object
`slot.use`	slot to use

showIntr *show method for CytoSignal*

Description

show method for CytoSignal

Usage

```
showIntr(object, slot.use = NULL, signif.use = NULL, return.name = FALSE)
```

Arguments

object	CytoSignal object
slot.use	slot to use
signif.use	Significance level to use for plotting
return.name	Whether to return interaction name that comes with the form of "ligand-receptor", instead of showing the unique IDs. Default FALSE.

Value

By default, character vector of unique IDs of the interactions available to the specified slots. When return.name = TRUE, a named character vector where the names are the IDs and the values are interaction names with the form of "ligand-receptor".

showLog *show all current logs*

Description

show all current logs

Usage

```
showLog(object)
```

Arguments

object	CytoSignal object

showScore *show method for lrScores*

Description

show method for lrScores

Usage

```
showScore(object, slot.use = NULL)
```

Arguments

object	CytoSignal object
slot.use	slot to use

showUnpt *show intr.data in ImpData*

Description

show intr.data in ImpData

Usage

```
showUnpt(object, slot.use = NULL)
```

Arguments

object	CytoSignal object
slot.use	slot to use

showVelo *show method for lrVelo*

Description

show method for lrVelo

Usage

```
showVelo(object, slot.use = NULL)
```

Arguments

object	CytoSignal object
slot.use	slot to use

suggestScaleFactor *Suggest scaling factor of real units to spatial units*

Description

This function is used to estimate the scaling facor of real units to spatial units. Returns the top 5 possible scaling factors, users can choose to which to use.

Usage

```
suggestScaleFactor(object, cell.intervel = NULL)
```

Arguments

object	CytoSignal object
cell.intervel	Distance between two cell in physical units

Value

a vector of possible scaling factors

uniqueColors *Generate character vector of unique color hex codes Base color palette adopts* ggsci:::ggsci_dbigvdefault

Description

Generate character vector of unique color hex codes Base color palette adopts ggsci:::ggsci_dbigvdefault

Usage

```
uniqueColors(n)
```

Arguments

n	Number of unique colors to generate. When not exceeding existing number of colors, just return the first n colors, otherwise generate interpolated colors.

Index