Package 'CiteFuse'

April 7, 2025

Type Package

Title CiteFuse: multi-modal analysis of CITE-seq data

Version 1.19.0

Description CiteFuse pacakage implements a suite of methods and tools for CITE-seq data from pre-processing to integrative analytics, including doublet detection, network-based modality integration, cell type clustering, differential RNA and protein expression analysis, ADT evaluation, ligand-receptor interaction analysis, and interactive web-based visualisation of the analyses.

License GPL-3

Encoding UTF-8

Depends R (>= 4.0)

Imports SingleCellExperiment (>= 1.8.0), SummarizedExperiment (>= 1.16.0), Matrix, mixtools, cowplot, ggplot2, gridExtra, grid, dbscan, uwot, Rtsne, S4Vectors (>= 0.24.0), igraph, scales, scran (>= 1.14.6), graphics, methods, stats, utils, reshape2, ggridges, randomForest, pheatmap, ggraph, grDevices, rhdf5, rlang, Rcpp, compositions

LinkingTo Rcpp

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, DT, mclust, scater, ExPosition, BiocStyle, pkgdown

VignetteBuilder knitr

LazyData false

biocViews SingleCell, GeneExpression

BugReports https://github.com/SydneyBioX/CiteFuse/issues

git_url https://git.bioconductor.org/packages/CiteFuse

git_branch devel

git_last_commit 699f98c

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2025-04-07

Author Yingxin Lin [aut, cre],
Hani Kim [aut]

Maintainer Yingxin Lin <yingxin.lin@sydney.edu.au>

Contents

`CiteFuse` *CiteFuse*

Description

A function to runSNF for CITE seq data

Usage

```
CiteFuse(
  sce,
  altExp_name = "ADT",
  W_list = NULL,
  gene_select = TRUE,
  dist_cal_RNA = "correlation",
  dist_cal_ADT = "propr",
  ADT_subset = NULL,
  K_knn = 20,
  K_knn_Aff = 30,
  sigma = 0.45,
  t = 10,
  metadata_names = NULL,
  verbose = TRUE,
  topN = 2000
)
```

Arguments

sce	a SingleCellExperiment
altExp_name	expression name of ADT matrix
W_list	affinity list, if it is NULL, the function will calculate it.
gene_select	whether highly variable genes will be selected for RNA-seq to calcualte simlarity matrix using 'scran' package
dist_cal_RNA	similarity metrics used for RNA matrix
dist_cal_ADT	similarity metrics used for ADT matrix
ADT_subset	A vector indicates the subset that will be used.
K_knn	Number of nearest neighbours
K_knn_Aff	Number of nearest neighbors for computing affinity matrix
sigma	Variance for local model for computing affinity matrix
t	Number of iterations for the diffusion process.
metadata_names	A vector indicates the names of metadata returned
verbose	whether print out the process
topN	top highly variable genes are used variable gene selection (see 'modelGeneVar' in 'scran' package for more details)

Value

A SingleCellExperiment object with fused matrix results stored

References

B Wang, A Mezlini, F Demir, M Fiume, T Zu, M Brudno, B Haibe-Kains, A Goldenberg (2014) Similarity Network Fusion: a fast and effective method to aggregate multiple data types on a genome wide scale. Nature Methods. Online. Jan 26, 2014

Examples

```
data("sce_ctcl_subset", package = "CiteFuse")
sce_ctcl_subset <- CiteFuse(sce_ctcl_subset)
```

`CITEseq_example` *A subset of ECCITE-seq data (control)*

Description

Data from Mimitou et al. ECCITE-seq PBMC control sample data, which is a list of three matrices of RNA, ADT and HTO

Usage

```
data(CITEseq_example, package = 'CiteFuse')
```

Format

An object of class `list` of length 3.

Source

Gene Expression Omnibus with the accession code GSE126310.

References

Mimitou, E. P., Cheng, A., Montalbano, A., et al. (2019). Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. Nature Methods, 16(5), 409–412.

`crossSampleDoublets` *crossSampleDoublets*

Description

A function that perform normalisation for alternative expression

Usage

```
crossSampleDoublets(sce, altExp_name = NULL, totalExp_threshold = 10)
```

Arguments

sce	A SingleCellExperiment object
altExp_name	Name of alternative expression that will be used to perform normalisation. If it is NULL, it will set to HTO.
totalExp_threshold	the threshold indicates for the HTO less than this threshold will be filtered from the analysis

Value

A SingleCellExperiment Object

Examples

```
data(CITEseq_example, package = "CiteFuse")
sce_citeseq <- preprocessing(CITEseq_example)
sce_citeseq <- normaliseExprs(sce = sce_citeseq,
altExp_name = "HTO",
transform = "log")
sce_citeseq <- crossSampleDoublets(sce_citeseq)
```

DEbubblePlot *DEbubblePlot*

Description

A function to generate circlepack plot to visualise the marker for each cluster

Usage

```
DEbubblePlot(de_list)
```

Arguments

de_list	A list of results from 'DE genes ()'

Value

A ggplot to visualise the DE results via bubble plot

Examples

```
library(S4Vectors)
data(sce_control_subset, package = "CiteFuse")
sce_control_subset <- DEgenes(sce_control_subset,
altExp_name = "none",
group = sce_control_subset$SNF_W_louvain,
return_all = TRUE,
exprs_pct = 0.5)


sce_control_subset <- selectDEgenes(sce_control_subset,
                            altExp_name = "none")

sce_control_subset <- DEgenes(sce_control_subset,
altExp_name = "ADT",
group = sce_control_subset$SNF_W_louvain,
return_all = TRUE,
exprs_pct = 0.5)


sce_control_subset <- selectDEgenes(sce_control_subset,
                             altExp_name = "ADT")

rna_DEgenes <- metadata(sce_control_subset)[["DE_res_RNA_filter"]]
adt_DEgenes <- metadata(sce_control_subset)[["DE_res_ADT_filter"]]

rna_DEgenes <- lapply(rna_DEgenes, function(x){
  x$name <- gsub("hg19_", "", x$name)
  x})
DEbubblePlot(list(RNA = rna_DEgenes, ADT = adt_DEgenes))
```

`DEcomparisonPlot` *DEcomparisonPlot*

Description

A function to visualise the pairwise comparison of pvalue in different data modality.

Usage

```
DEcomparisonPlot(de_list, feature_list)
```

Arguments

`de_list`	A list including two lists results from 'DE genes ()'.
`feature_list`	A list including two lists features indicating the selected subset of features will be visualised

Value

A ggplot2 to visualise the comparison plot of DE.

Examples

```
library(S4Vectors)
data(sce_control_subset)
sce_control_subset <- DEgenes(sce_control_subset,
group = sce_control_subset$SNF_W_louvain,
return_all = TRUE,
exprs_pct = 0.5)

sce_control_subset <- selectDEgenes(sce_control_subset)

sce_control_subset <- DEgenes(sce_control_subset,
                       altExp_name = "ADT",
                       group = sce_control_subset$SNF_W_louvain,
                       return_all = TRUE,
                       exprs_pct = 0.5)


sce_control_subset <- selectDEgenes(sce_control_subset,
                             altExp_name = "ADT")
rna_list <- c("hg19_CD4",
"hg19_CD8A",
"hg19_HLA-DRB1",
"hg19_ITGAX",
"hg19_NCAM1",
"hg19_CD27",
"hg19_CD19")

adt_list <- c("CD4", "CD8", "MHCII (HLA-DR)", "CD11c", "CD56", "CD27", "CD19")

rna_DEgenes_all <- S4Vectors::metadata(sce_control_subset)[["DE_res_RNA"]]
adt_DEgenes_all <- S4Vectors::metadata(sce_control_subset)[["DE_res_ADT"]]

feature_list <- list(RNA = rna_list, ADT = adt_list)
de_list <- list(RNA = rna_DEgenes_all, ADT = adt_DEgenes_all)

DEcomparisonPlot(de_list = de_list,
                 feature_list = feature_list)
```

DEgenes *DEgenes*

Description

A function to perform DE analysis on CITE seq data

Usage

```
DEgenes(
  sce,
  altExp_name = "none",
  exprs_value = "logcounts",
  group = NULL,
  method = "wilcox",
  exprs_pct = 0.1,
  exprs_threshold = 0,
  return_all = FALSE,
  pval_adj = 0.05,
  mean_diff = 0,
  pct_diff = 0.1,
  topN = 10
)
```

Arguments

sce	A SingleCellExperiment object
altExp_name	A character indicates which expression matrix is used. by default is none (i.e. RNA).
exprs_value	A character indicates which expression value in assayNames is used.
group	A vector indicates the grouping of the data
method	A character indicates the method used in DE analysis
exprs_pct	A numeric indicates the threshold expression percentage of a gene to be considered in DE analysis
exprs_threshold	A numeric indicates the threshold of expression. By default is 0.
return_all	Whether return full list of DE genes
pval_adj	A numeric indicates the threshold of adjusted p-value.
mean_diff	A numeric indicates the threshold of difference of average expression.
pct_diff	A numeric indicates the threshold of difference of percentage expression.
topN	A numeric indicates the top number of genes will be included in the list.

Value

A SingleCellExeperiment with DE results stored in meta data DE_res

Examples

```
data(sce_control_subset)
sce_control_subset <- DEgenes(sce_control_subset,
group = sce_control_subset$SNF_W_louvain,
return_all = TRUE,
exprs_pct = 0.5)
```

```
sce_control_subset <- selectDEgenes(sce_control_subset)
```

DEgenesCross *DEgenesCross*

Description

A function to perform DE analysis on a list of CITE seq data

Usage

```
DEgenesCross(
  sce_list,
  altExp_name = "none",
  exprs_value = "logcounts",
  method = "wilcox",
  colData_name = NULL,
  group_to_test = NULL,
  exprs_pct = 0.1,
  exprs_threshold = 0,
  return_all = FALSE,
  pval_adj = 0.05,
  mean_diff = 0,
  pct_diff = 0.1,
  topN = 10
)
```

Arguments

sce_list	A Slist of ingleCellExperiment object
altExp_name	A character indicates which expression matrix is used. by default is none (i.e. RNA).
exprs_value	A character indicates which expression value in assayNames is used.
method	A character indicates the method used in DE analysis
colData_name	A vector of character indicates the colData that stored the group information of each sce of the sce_list
group_to_test	A vector of character indicates which group in each sce is used to compared across the sce list.
exprs_pct	A numeric indicates the threshold expression percentage of a gene to be considered in DE analysis
exprs_threshold	A numeric indicates the threshold of expression. By default is 0.
return_all	Whether return full list of DE genes
pval_adj	A numeric indicates the threshold of adjusted p-value.

mean_diff	A numeric indicates the threshold of difference of average expression.
pct_diff	A numeric indicates the threshold of difference of percentage expression.
topN	A numeric indicates the top number of genes will be included in the list.

Value

A SingleCellExeperiment with DE results stored in meta data DE_res

Examples

```
data("sce_control_subset", package = "CiteFuse")
data("sce_ctcl_subset", package = "CiteFuse")

de_res <- DEgenesCross(sce_list = list(control = sce_control_subset,
ctcl = sce_ctcl_subset),
colData_name = c("SNF_W_louvain", "SNF_W_louvain"),
group_to_test = c("2", "6"))
```

---

geneADTnetwork *geneADTnetwork*

Description

A function to visualise the features distribtuion

Usage

```
geneADTnetwork(
  sce,
  RNA_exprs_value = "logcounts",
  altExp_name = "ADT",
  altExp_exprs_value = "logcounts",
  RNA_feature_subset = NULL,
  ADT_feature_subset = NULL,
  cell_subset = NULL,
  cor_threshold = 0.5,
  cor_method = c("pearson", "kendall", "spearman"),
  RNA_exprs_pct = 0.1,
  ADT_exprs_pct = 0.1,
  RNA_exprs_threshold = 0,
  ADT_exprs_threshold = 0,
  network_layout = NULL,
  return_igraph = FALSE
)
```

Arguments

sce	A singlecellexperiment object
RNA_exprs_value	A character indicates which expression value for RNA in assayNames is used.
altExp_name	A character indicates which expression matrix is used. by default is none (i.e. RNA).
altExp_exprs_value	A character indicates which expression value in assayNames is used.
RNA_feature_subset	A vector of characters indicates the subset of features of RNA that are used for visualisation
ADT_feature_subset	A vector of characters indicates the subset of features of ADT that are used for visualisation
cell_subset	A vector of characters indicates the subset of cells that are used for visualisation
cor_threshold	Thresholds of correlation.
cor_method	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated.
RNA_exprs_pct	A numeric indicates the threshold expression percentage of a gene to be considered in correlation analysis
ADT_exprs_pct	A numeric indicates the threshold expression percentage of a gene to be considered in correlation analysis
RNA_exprs_threshold	A numeric indicates the threshold of RNA expression. By default is 0.
ADT_exprs_threshold	A numeric indicates the threshold of ADT expression. By default is 0.
network_layout	layout of the network
return_igraph	indicates whether return the igraph object

Value

A igraph object of gene-ADT network

Examples

```
library(SingleCellExperiment)
set.seed(2020)
data(sce_control_subset, package = "CiteFuse")
RNA_feature_subset <- sample(rownames(sce_control_subset), 50)
ADT_feature_subset <- rownames(altExp(sce_control_subset, "ADT"))

geneADTnetwork(sce_control_subset,
               RNA_feature_subset = RNA_feature_subset,
               ADT_feature_subset = ADT_feature_subset,
               cor_method = "pearson",
```

```
                network_layout = igraph::layout_with_fr)
```

`igraphClustering` *igraphClustering*

Description

A function to perform igraph clustering

Usage

```
igraphClustering(
  sce,
  metadata = "SNF_W",
 method = c("louvain", "leiden", "walktrap", "spinglass", "optimal", "leading_eigen",
    "label_prop", "fast_greedy", "edge_betweenness"),
  ...
)
```

Arguments

`sce`	A singlecellexperiment object
`metadata`	indicates the meta data name of affinity matrix to virsualise
`method`	A character indicates the method for finding communities from igraph. Default is louvain clustering.
`...`	Other inputs for the igraph functions

Value

A vector indicates the membership (clustering) results

Examples

```
data(sce_control_subset, package = "CiteFuse")
sce_control_subset <- CiteFuse(sce_control_subset)
SNF_W_louvain <- igraphClustering(sce_control_subset,
method = "louvain")
```

importanceADT *importanceADT*

Description

A function to calculate the importance score of ADT

Usage

```
importanceADT(
  sce,
  altExp_name = "ADT",
  exprs_value = "logcounts",
  method = c("randomForest", "PCA"),
  group = NULL,
  subsample = TRUE,
  times = 10,
  prop = 0.8,
  k_pca = 5,
  remove_first_PC = TRUE,
  ...
)
```

Arguments

sce	A singlecellexperiment object
altExp_name	A character indicates which expression matrix is used. by default is none (i.e. RNA).
exprs_value	A character indicates which expression value in assayNames is used.
method	A character indicates the method of ADT importance calculation, either randomForest or PCA
group	A vector indicates the grouping of the data (for random forest)
subsample	Whether perform subsampling (for random forest)
times	A numeric indicates the times of subsampling is performed (for random forest)
prop	A numeric indicates the proportion of cells are subsampled from the whole data (for random forest)
k_pca	Number of principal component will be used to calculate the loading scores (for PCA)
remove_first_PC	A logical input indicates whether the first component will be removed from calculation (for PCA).
...	other arguments to ‘randomForest()‘ or ‘prcomp()‘ function

Details

For random forest, the importance scores are based on features importance. For PCA, it implements the method proposed in Levin et al (based on the loading of features).

Value

A SingleCellExperiment object

References

Levine, J.H., Simonds, E.F., Bendall, S.C., Davis, K.L., El-ad, D.A., Tadmor, M.D., Litvin, O., Fienberg, H.G., Jager, A., Zunder, E.R. and Finck, R., 2015. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell, 162(1), pp.184-197.

Examples

```
data("sce_control_subset", package = "CiteFuse")
sce_control_subset <- importanceADT(sce_control_subset,
group = sce_control_subset$SNF_W_louvain,
subsample = TRUE)
```

---

ligandReceptorTest *ligandReceptorTest*

Description

A function to perform ligand receptor analysis

Usage

```
ligandReceptorTest(
  sce,
  ligandReceptor_list,
  cluster,
  RNA_exprs_value = "minMax",
  use_alt_exp = TRUE,
  altExp_name = "ADT",
  altExp_exprs_value = "zi_minMax",
  num_permute = 1000,
  p_sig = 0.05
)
```

Arguments

sce	A singlecellexperiment object
ligandReceptor_list	A data.frame indicates the ligand receptor list
cluster	A vector indicates the cluster results
RNA_exprs_value	A character indicates which expression value for RNA in assayNames is used.
use_alt_exp	A logical vector indicates whether receptors expression will use alternative expression matrix to quantify.
altExp_name	A character indicates which expression matrix is used. by default is ADT .
altExp_exprs_value	A character indicates which expression value in assayNames is used.
num_permute	Number of permutation.
p_sig	A numeric indicates threshold of the pvalue significance

Value

A SingleCellExperiment object with ligand receptor results

Examples

```
data(lr_pair_subset, package = "CiteFuse")
data(sce_control_subset, package = "CiteFuse")

sce_control_subset <- normaliseExprs(sce = sce_control_subset,
altExp_name = "ADT",
transform = "zi_minMax")

sce_control_subset <- normaliseExprs(sce = sce_control_subset,
                             altExp_name = "none",
                             exprs_value = "logcounts",
                             transform = "minMax")

sce_control_subset <- ligandReceptorTest(sce = sce_control_subset,
                                 ligandReceptor_list = lr_pair_subset,
                                 cluster = sce_control_subset$SNF_W_louvain,
                                 RNA_exprs_value = "minMax",
                                 use_alt_exp = TRUE,
                                 altExp_name = "ADT",
                                 altExp_exprs_value = "zi_minMax",
                                 num_permute = 100)
```

lr_pair_subset *A subset of Ligand Receptor Pairs*

Description

A subset of Ligand Receptor Pairs

Usage

```
data(lr_pair_subset, package = 'CiteFuse')
```

Format

An object of class `matrix` (inherits from `array`) with 50 rows and 2 columns.

normaliseExprs *normaliseExprs*

Description

A function that perform normalisation for alternative expression

Usage

```
normaliseExprs(
  sce,
  altExp_name = NULL,
  exprs_value = "counts",
  transform = c("log", "clr", "zi_minMax", "minMax"),
  log_offset = NULL
)
```

Arguments

`sce`	A `SingleCellExperiment` object
`altExp_name`	Name of alternative expression that will be used to perform normalisation
`exprs_value`	A character indicates which expression value in assayNames is used.
`transform`	type of transformation, either log or clr (Centered log ratio transform)
`log_offset`	Numeric scalar specifying the pseudo-count to add when log-transforming expression values. Default is 1

Value

a SingleCellExperiment object

Examples

```
data(CITEseq_example, package = "CiteFuse")
sce_citeseq <- preprocessing(CITEseq_example)
sce_citeseq <- normaliseExprs(sce = sce_citeseq,
altExp_name = "ADT",
transform = "log")
```

---

plotHTO *plotHTO*

Description

A function to plot HTO expression

Usage

```
plotHTO(sce, which_idx = seq_len(2), altExp_name = NULL, ncol = 2)
```

Arguments

sce	sce
which_idx	which_idx
altExp_name	altExp_name
ncol	ncol

Value

A plot visualising the HTO expression

Examples

```
data(CITEseq_example, package = "CiteFuse")
sce_citeseq <- preprocessing(CITEseq_example)
sce_citeseq <- normaliseExprs(sce = sce_citeseq,
altExp_name = "HTO",
transform = "log")
plotHTO(sce_citeseq, 1:4)
```

plotHTOSingle *plotHTOSingle*

Description

A function to plot HTO expression

Usage

```
plotHTOSingle(sce, which_idx = seq_len(2), altExp_name = NULL)
```

Arguments

sce	sce
which_idx	which_idx
altExp_name	altExp_name

Value

A plot visualising the HTO expression

preprocessing *A function to preprocess the list of expression matrix*

Description

This function will keep the samples that are common across the list of expression matrix, and filter the features that are all zeros across samples, and finally construct a `SingleCellExperiment` object

Usage

```
preprocessing(
  exprsMat = NULL,
  return_sce = TRUE,
  assay_matrix = 1,
  filter_features = TRUE,
  rowData = NULL,
  colData = NULL
)
```

Arguments

exprsMat	A list or a matrix indicates the expression matrices of the testing datasets (each matrix must be matrix or dgCMatrix class)
return_sce	A logical input indicates whether a SingleCellExperiment object will be return
assay_matrix	A integer indicates which list will be used as 'assay' input of 'SingleCellExperiment'
filter_features	A logical input indicates whether the features with all zeros will be removed
rowData	A DataFrame indicates the rowData to be stored in the sce object
colData	A DataFrame indicates the colData to be stored in the sce object

Value

either a SingleCellExperiment object or a preprocessed expression matrix

Examples

```
data(CITEseq_example, package = "CiteFuse")
sce_citeseq <- preprocessing(CITEseq_example)
```

---

readFrom10X *readFrom10X*

Description

A function to read the data from 10X

Usage

```
readFrom10X(
  dir,
  type = c("auto", "sparse", "HDF5"),
  feature_named_by = c("gene_id", "gene_symbol"),
  filter_features = TRUE
)
```

Arguments

dir	A character indicates the directory of the 10X files
type	A character indicates the format of the data, sparse or HDF5
feature_named_by	A character indicates whehter the genes will be named by gene_id or gene_symbol
filter_features	A logical input indicates whether the features with all zeros will be removed

Value

a SingleCellExperiment object

Examples

```
## Not run:
tmpdir <- tempdir()
tenXdata <- "http://cf.10xgenomics.com/samples/cell-exp/3.1.0/connect_5k_pbmc_NGSC3_ch1/"
file <- "connect_5k_pbmc_NGSC3_ch1_filtered_feature_bc_matrix.tar.gz"
download.file(paste0(tenXdata, file),file.path(tmpdir, file))
untar(file.path(tmpdir,file),
      exdir = tmpdir)
sce_citeseq_10X <- readFrom10X(file.path(tmpdir,
"filtered_feature_bc_matrix/"))
sce_citeseq_10X

## End(Not run)
```

`reducedDimSNF` *reducedDimSNF*

Description

A function to reduce the dimension of the similarity matrix

Usage

```
reducedDimSNF(sce, metadata = "SNF_W", method = "UMAP", dimNames = NULL, ...)
```

Arguments

`sce`	A singlecellexperiment object
`metadata`	indicates the meta data name of affinity matrix to virsualise
`method`	the method of visualisation, which can be UMAP, tSNE and diffusion map
`dimNames`	indicates the name of the reduced dimension results.
`...`	other parameters for tsne(), umap()

Value

A SingleCellExperiment object

Examples

```
data(sce_control_subset, package = "CiteFuse")
sce_control_subset <- CiteFuse(sce_control_subset)
sce_control_subset <- reducedDimSNF(sce_control_subset,
method = "tSNE",
dimNames = "tSNE_joint")
```

---

sce_control_subset *A SingleCellExperiment of ECCITE-seq data*

Description

Data from Mimitou et al. ECCITE-seq PBMC Control sample data

Usage

```
data(sce_control_subset, package = 'CiteFuse')
```

Format

An object of class `SingleCellExperiment` with 1508 rows and 128 columns.

Source

Gene Expression Omnibus with the accession code GSE126310.

References

Mimitou, E. P., Cheng, A., Montalbano, A., et al. (2019). Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. Nature Methods, 16(5), 409–412.

---

sce_ctcl_subset *A SingleCellExperiment of ECCITE-seq data*

Description

Data from Mimitou et al. ECCITE-seq PBMC CTCL sample data

Usage

```
data(sce_ctcl_subset, package = 'CiteFuse')
```

Format

An object of class `SingleCellExperiment` with 1450 rows and 173 columns.

Source

Gene Expression Omnibus with the accession code GSE126310.

References

Mimitou, E. P., Cheng, A., Montalbano, A., et al. (2019). Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. Nature Methods, 16(5), 409–412.

`selectDEgenes` *selectDEgenes*

Description

A function to select DE genes

Usage

```
selectDEgenes(
  sce = NULL,
  de_res = NULL,
  altExp_name = "none",
  pval_adj = 0.05,
  mean_diff = 0,
  pct_diff = 0.1,
  topN = 10
)
```

Arguments

`sce`	A SingleCellExperiment object with DE results stored in meta data DE_res list.
`de_res`	DE_res returned by DEgenesCross().
`altExp_name`	A character indicates which expression matrix is used. by default is none (i.e. RNA).
`pval_adj`	A numeric indicates the threshold of adjusted p-value.
`mean_diff`	A numeric indicates the threshold of difference of average expression.
`pct_diff`	A numeric indicates the threshold of difference of percentage expression.
`topN`	A numeric indicates the top number of genes will be included in the list.

Value

A SingleCellExperiment With filtered DE results in DE_res_filter list of metadata

Examples

```
data(sce_control_subset)
sce_control_subset <- DEgenes(sce_control_subset,
group = sce_control_subset$SNF_W_louvain,
return_all = TRUE,
exprs_pct = 0.5)

sce_control_subset <- selectDEgenes(sce_control_subset)
```

`spectralClustering` *spectralClustering*

Description

A function to perform spectral clustering

Usage

```
spectralClustering(affinity, K = 20, delta = 1e-05)
```

Arguments

`affinity`	An affinity matrix
`K`	number of clusters
`delta`	delta

Value

A list indicates the spectral clustering results

Examples

```
data(sce_control_subset, package = "CiteFuse")
sce_control_subset <- CiteFuse(sce_control_subset)
SNF_W <- S4Vectors::metadata(sce_control_subset)[["SNF_W"]]
SNF_W_clust <- spectralClustering(SNF_W, K = 5)
```

visImportance *visImportance*

Description

A function to visualise the features distribtuion

Usage

```
visImportance(
  sce,
  plot = c("boxplot", "heatmap"),
  altExp_name = "ADT",
  exprs_value = "logcounts"
)
```

Arguments

sce	A singlecellexperiment object
plot	A string indicates the type of the plot (either boxplot or heatmap)
altExp_name	A character indicates which expression matrix is used. by default is none (i.e. RNA).
exprs_value	A character indicates which expression value in assayNames is used.

Value

A plot (either ggplot or pheatmap) to visualise the ADT importance results

Examples

```
data("sce_control_subset", package = "CiteFuse")
sce_control_subset <- importanceADT(sce_control_subset,
group = sce_control_subset$SNF_W_louvain,
subsample = TRUE)
visImportance(sce_control_subset, plot = "boxplot")
```

visLigandReceptor *visLigandReceptor*

Description

A function to visualise ligand receptor analysis

Usage

```
visLigandReceptor(
  sce,
  type = c("pval_heatmap", "pval_dotplot", "group_network", "group_heatmap",
    "lr_network"),
  receptor_type = NULL
)
```

Arguments

sce	A singlecellexperiment object
type	A character indicates the type of the plot for ligand receptor restuls visualisation, option includes "pval_heatmap", "pval_dotplot", "group_network", "group_heatmap", and "lr_network"
receptor_type	A character indicates which receptor expression's ligand receptor results are used to generate the figures.

Value

A plot visualise the ligand receptor results

Examples

```
data(lr_pair_subset, package = "CiteFuse")
data(sce_control_subset, package = "CiteFuse")

sce_control_subset <- normaliseExprs(sce = sce_control_subset,
altExp_name = "ADT",
transform = "zi_minMax")

sce_control_subset <- normaliseExprs(sce = sce_control_subset,
                             altExp_name = "none",
                             exprs_value = "logcounts",
                             transform = "minMax")

sce_control_subset <- ligandReceptorTest(sce = sce_control_subset,
                                 ligandReceptor_list = lr_pair_subset,
                                 cluster = sce_control_subset$SNF_W_louvain,
                                 RNA_exprs_value = "minMax",
                                 use_alt_exp = TRUE,
```

```
                                altExp_name = "ADT",
                                altExp_exprs_value = "zi_minMax",
                                num_permute = 100)
visLigandReceptor(sce_control_subset,
type = "pval_heatmap",
receptor_type = "ADT")
```

---

visualiseDim *visualiseDim*

Description

A function to visualise the reduced dimension

Usage

```
visualiseDim(
  sce,
  dimNames = NULL,
  colour_by = NULL,
  shape_by = NULL,
  data_from = c("colData", "assay", "altExp"),
  assay_name = NULL,
  altExp_name = NULL,
  altExp_assay_name = NULL,
  dim = seq_len(2)
)
```

Arguments

sce	A singlecellexperiment object
dimNames	indicates the name of the reduced dimension results.
colour_by	A character indicates how the cells coloured by. The information either stored in colData, assay, or altExp.
shape_by	A character indicates how the cells shaped by. The information either stored in colData, assay, or altExp.
data_from	A character indicates where the colour by data stored
assay_name	A character indicates the assay name of the expression
altExp_name	A character indicates the name of alternative expression
altExp_assay_name	A character indicates the assay name of alternative expression
dim	a vector of numeric with length of 2 indicates which component is being plot

Value

A ggplot of the reduced dimension visualisation

Examples

```
data(sce_control_subset, package = "CiteFuse")
sce_control_subset <- CiteFuse(sce_control_subset)
sce_control_subset <- reducedDimSNF(sce_control_subset,
method = "tSNE",
dimNames = "tSNE_joint")
visualiseDim(sce_control_subset, dimNames = "tSNE_joint",
colour_by = "SNF_W_clust")
```

`visualiseExprs` *visualiseExprs*

Description

A function to visualise the features distribtuion

Usage

```
visualiseExprs(
  sce,
  plot = c("boxplot", "violin", "jitter", "density", "pairwise"),
  altExp_name = c("none"),
  exprs_value = "logcounts",
  group_by = NULL,
  facet_by = NULL,
  feature_subset = NULL,
  cell_subset = NULL,
  n = NULL,
  threshold = NULL
)
```

Arguments

`sce`	A singlecellexperiment object
`plot`	Type of plot, includes boxplot, violin, jitter, density, and pairwise. By default is boxplot
`altExp_name`	A character indicates which expression matrix is used. by default is none (i.e. RNA).
`exprs_value`	A character indicates which expression value in assayNames is used.
`group_by`	A character indicates how is the expression will be group in the plots (stored in colData).
`facet_by`	A character indicates how is the expression will be lay out panels in a grid in the plots (stored in colData).
`feature_subset`	A vector of characters indicates the subset of features that are used for visualisation

cell_subset	A vector of characters indicates the subset of cells that are used for visualisation
n	A numeric indicates the top expressed features to show.
threshold	Thresholds of high expresion for features (only is used for pairwise plot).

Value

A ggplot to visualise te features distribution

Examples

```
data(sce_control_subset)
visualiseExprs(sce_control_subset,
plot = "boxplot",
group_by = "SNF_W_louvain",
feature_subset = c("hg19_CD8A"))

visualiseExprs(sce_control_subset,
plot = "density",
altExp_name = "ADT",
group_by = "SNF_W_louvain",
feature_subset = c("CD8", "CD4"))
```

---

visualiseExprsList *visualiseExprsList*

Description

A function to visualise the features distribtuion for a list of SingleCellExperiment

Usage

```
visualiseExprsList(
  sce_list,
  plot = c("boxplot", "violin", "jitter", "density"),
  altExp_name = "none",
  exprs_value = "logcounts",
  group_by = NULL,
  feature_subset = NULL,
  cell_subset = NULL,
  n = NULL
)
```

Arguments

sce_list	A list of SingleCellExperiment object
plot	Type of plot, includes boxplot, violin, jitter, density, and pairwise. By default is boxplot
altExp_name	A character indicates which expression matrix is used. by default is none (i.e. RNA).
exprs_value	A character indicates which expression value in assayNames is used.
group_by	A character indicates how is the expression will be group in the plots (stored in colData).
feature_subset	A vector of characters indicates the subset of features that are used for visualisation
cell_subset	A vector of characters indicates the subset of cells that are used for visualisation
n	A numeric indicates the top expressed features to show.

Value

A ggplot to visualise te features distribution

Examples

```
data(sce_control_subset, package = "CiteFuse")
data(sce_ctcl_subset, package = "CiteFuse")
visualiseExprsList(sce_list = list(control = sce_control_subset,
ctcl = sce_ctcl_subset),
plot = "boxplot",
altExp_name = "none",
exprs_value = "logcounts",
feature_subset = c("hg19_CD8A"),
group_by = c("SNF_W_louvain", "SNF_W_louvain"))
```

---

visualiseKNN *visualiseKNN*

Description

A function to perform louvain clustering

Usage

```
visualiseKNN(sce, colour_by = NULL, metadata = "SNF_W")
```

Arguments

sce	A singlecellexperiment object
colour_by	the name of coldata that is used to colour the node
metadata	indicates the meta data name of affinity matrix to virsualise

Value

A igraph plot

Examples

```
data(sce_control_subset, package = "CiteFuse")
sce_control_subset <- CiteFuse(sce_control_subset)
SNF_W_louvain <- igraphClustering(sce_control_subset,
method = "louvain")
visualiseKNN(sce_control_subset, colour_by = "SNF_W_louvain")
```

---

withinSampleDoublets *withinSampleDoublets*

Description

doublet identification within batch

Usage

```
withinSampleDoublets(sce, altExp_name = NULL, eps = 200, minPts = 50)
```

Arguments

sce	a SingleCellExperiment
altExp_name	expression name of HTO matrix
eps	eps of DBSCAN
minPts	minPts of DBSCAN

Value

A SingleCellExperiment object

Examples

```
data(CITEseq_example, package = "CiteFuse")
sce_citeseq <- preprocessing(CITEseq_example)
sce_citeseq <- normaliseExprs(sce = sce_citeseq,
altExp_name = "HTO",
transform = "log")
sce_citeseq <- crossSampleDoublets(sce_citeseq)
sce_citeseq <- withinSampleDoublets(sce_citeseq,
minPts = 10)
```

Index