

Package ‘DR.SC’

October 3, 2025

Type Package

Title Joint Dimension Reduction and Spatial Clustering

Version 3.6

Date 2025-10-02

Author Wei Liu [aut, cre],
Yi Yang [aut],
Jin Liu [aut]

Maintainer Wei Liu <liuweideng@gmail.com>

Description Joint dimension reduction and spatial clustering is conducted for Single-cell RNA sequencing and spatial transcriptomics data, and more details can be referred to Wei Liu, Xu Liao, Yi Yang, Huazhen Lin, Joe Yeong, Xiang Zhou, Xingjie Shi and Jin Liu. (2022) <[doi:10.1093/nar/gkac219](https://doi.org/10.1093/nar/gkac219)>. It is not only computationally efficient and scalable to the sample size increment, but also is capable of choosing the smoothness parameter and the number of clusters as well.

License GPL-3

Depends parallel, spatstat.geom, R (>= 4.0.0)

Imports CompQuadForm, irlba, cowplot, ggplot2, MASS, Matrix, mclust, methods, purrr, S4Vectors, RColorBrewer, Rcpp (>= 1.0.5), Seurat, stats

Suggests knitr, rmarkdown

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation yes

RoxygenNote 7.1.1

URL <https://github.com/feiyong/DR.SC>

BugReports <https://github.com/feiyong/DR.SC/issues>

LazyData true

Repository CRAN

Date/Publication 2025-10-03 05:20:34 UTC

Contents

dlpfc151510	2
DR.SC	3
DR.SC_fit	5
drscPlot	7
FindSVGs	8
getAdj	10
getAdj_auto	11
getAdj_manual	12
getneighborhood_fast	12
mbicPlot	13
read10XVisium	14
readscRNAseq	15
RunWPCA	16
selectModel	18
seu	19
spatialPlotClusters	20
sp_means_Rcpp	21
sp_sums_Rcpp	21
topSVGs	22
Index	23

dlpfc151510	<i>A human dorsolateral prefrontal cortex data</i>
-------------	--

Description

A human dorsolateral prefrontal cortex dataset measured on the Visium platform, which invcludes 4634 spots and 500 genes, a subset of raw dataset at <https://github.com/LieberInstitute/spatialLIBD>.

Note

nothing

Author(s)

Wei Liu

References

None

Examples

```
data("dlpfc151510")
```

Description

Joint dimension reduction and spatial clustering for scRNA-seq and spatial transcriptomics data

Usage

```
## S3 method for class 'Seurat'
DR.SC(seu, K, q=15, platform= c('Visium', "ST", "Other_SRT", "scRNAseq"),...)
```

Arguments

seu	an object of class "Seurat". The details of this object are given under 'Details'.
q	a positive integer, specify the number of latent features to be extracted, default as 15.
K	a positive integer or integer vector, specify the number of clusters. When K is a vector, it is automatically selected by MBIC criteria. Users can also use BIC and AIC to select K or refine model with MBIC by argument <code>pen.const</code> in the function selectModel .
platform	a string, specify the platform of the provided data, default as "Visium". There are more platforms to be chosen, including ("Visium", "ST", "Other_SRT") and "scRNAseq", where the first group means there are spatial coordinates information in the metadata of seu, named "row" and "col" and a Hidden Markov random field is used to model the unobserved class label using spatial coordinates ("Other_SRT" represents the other SRT platforms except for 'Visium' and 'ST'), and the other group "scRNAseq" means there is no spatial information in object seu and a multinomial model is used to model the unobserved class labels. The platform helps to calculate the adjacency matrix.
...	Other arguments to pass into DR.SC_fit function.

Details

seu is an object named Seurat, thich can easily created by R package Seurat. DR-SC model can be applied to analyze both single cell RNA sequencing data and spatially resoved transcriptomics (SRT) data.

If the data are collected by the single cell RNA sequencing techonologies which means there is no spatial information in object seu then a multinomial model is used to model the unobserved class labels.

If the data is collected by the spatial transcriptomics technologies, then there are spatial coordinates information in the metadata of seu, named "row" and "col". DR-SC model uses a Hidden Markov random field to model the spatial coordinates. DR.SC supports different platforms of SRT data, such as 'Visium', 'ST' and any other platforms 'Other_SRT'.

For lattice grids in ST platform (ST), the interior spot has four neighbors (left, right, up and down), the boundary spot has three neighbors, and the spot in the corner has two neighbors. For hexagon grids, such as spatial coordinate in 10X Visium platform (Visium), the interior spot has six neighbors. For the irregular coordinates in other platforms (Other_SRT), Euclidean distance is adopted to decide whether a spot is a neighbor of another spot. For example, if the Euclidean distance between spot A and B is less than a radius, then A is taken as the neighbourhood of B. See function `getAdj` for more details.

Value

DR.SC returns a revised Seurat object. There are two revisions in the seu: 1. the metadata is added a new column named `spatial.drsc.cluster` that represents the clustering results from DR-SC model, and the `Idents(seu)` is assigned with `spatial.drsc.cluster`. 2. a DimReduc object named `dr-sc` is added in the slot reductions, which represents the features extracted by DR-SC model.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Xu Liao, Yi Yang, Huazhen Lin, Joe Yeong, Xiang Zhou, Xingjie Shi & Jin Liu (2022). Joint dimension reduction and clustering analysis of single-cell RNA-seq and spatial transcriptomics data, *Nucleic Acids Research*, gkac219.

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
data(seu)
library(Seurat)
seu <- NormalizeData(seu, verbose=FALSE)
# choose 100 highly variable features
# seu <- FindVariableFeatures(seu, nfeatures = 100)
# maxIter = 2 is only used for illustration, and user can use default.
# seu1 <- DR.SC(seu, K=4, platform = 'ST', maxIter=2, verbose=FALSE)

# choose spatially variable features (SVGs)
seu <- FindSVGs(seu, nfeatures = 40, verbose=FALSE)
# use SVGs to fit DR.SC model
# maxIter = 2 is only used for illustration, and user can use default.
seu1 <- DR.SC(seu, K=4, platform = 'ST', maxIter=2, verbose=TRUE)
```

DR.SC_fit

*Joint dimension reduction and spatial clustering***Description**

Joint dimension reduction and spatial clustering for scRNA-seq and spatial transcriptomics data

Usage

```
DR.SC_fit(X, K, Adj_sp=NULL, q=15,
          error.heter= TRUE, beta_grid=seq(0.5, 5, by=0.5),
          maxIter=25, epsLogLik=1e-5, verbose=FALSE, maxIter_ICM=6,
          wpca.int=FALSE, int.model="EEE", approxPCA=FALSE, coreNum = 5)
```

Arguments

X	a sparse matrix with class dgCMatrix or matrix, specify the log-normalization gene expression matrix used for DR-SC model.
K	a positive integer allowing scalar or vector, specify the number of clusters in model fitting.
Adj_sp	an optional sparse matrix with class dgCMatrix, specify the adjoint matrix used for DR-SC model. We provide this interface for those users who would like to define the adjacency matrix by their own.
q	a positive integer, specify the number of latent features to be extracted, default as 15. Usually, the choice of q is a trade-off between model complexity and fit to the data, and depends on the goals of the analysis and the structure of the data. A higher value will result in a more complex model with a higher number of parameters, which may lead to overfitting and poor generalization performance. On the other hand, a lower value will result in a simpler model with fewer parameters, but may also lead to underfitting and a poorer fit to the data.
error.heter	an optional logical value, whether use the heterogenous error for DR-SC model, default as TRUE. If error.heter=FALSE, then the homogenous error is used for probabilistic PCA model in DR-SC.
beta_grid	an optional vector of positive value, the candidate set of the smoothing parameter to be searched by the grid-search optimization approach.
maxIter	an optional positive value, represents the maximum iterations of EM.
epsLogLik	an optional positive vlaue, tolerance vlaue of relative variation rate of the observed pseudo log-loglikelihood value, default as '1e-5'.
verbose	an optional logical value, whether output the information of the ICM-EM algorithm.
maxIter_ICM	an optional positive value, represents the maximum iterations of ICM.
wpca.int	an optional logical value, means whether use the weighted PCA to obtain the initial values of loadings and other paramters, default as FALSE which means the ordinary PCA is used.

int.model	an optional string, specify which Gaussian mixture model is used in evaluating the initial values for DR-SC, default as "EEE"; and see Mclust for more models' names.
approxPCA	an optional logical value, whether use approximated PCA to speed up the computation for initial values.
coreNum	an optional positive integer, means the number of thread used in parallel computing, default as 5. If the length of K is one, then coreNum will be set as 1 automatically.

Details

Nothing

Value

DR.SC_fit returns a [list](#) with class "drscObject" with the following three components:

Objdrsc	a list including the model fitting results, in which the number of elements is same as the length of K.
out_param	a numeric matrix used for model selection in MBIC.
K_set	a scalar or vector equal to input argument K.

In addition, each element of "Objdrsc" is a list with the following components:

cluster	inferred class labels
hZ	extracted latent features.
beta	estimated smoothing parameter
Mu	mean vectors of mixtures components.
Sigma	covariance matrix of mixtures components.
W	estimated loading matrix
Lam_vec	estimated variance of errors in probabilistic PCA model
loglik	pseudo observed log-likelihood.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Xu Liao, Yi Yang, Huazhen Lin, Joe Yeong, Xiang Zhou, Xingjie Shi & Jin Liu (2022). Joint dimension reduction and clustering analysis of single-cell RNA-seq and spatial transcriptomics data, *Nucleic Acids Research*, gkac219.

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
data(seu)
library(Seurat)
seu <- NormalizeData(seu, verbose=FALSE)
# choose 40 highly variable features using FindVariableFeatures in Seurat
# seu <- FindVariableFeatures(seu, nfeatures = 40)
# or choose 40 spatailly variable features using FindSVGs in DR.SC
seu <- FindSVGs(seu, nfeatures = 40, verbose=FALSE)
# users define the adjacency matrix
Adj_sp <- getAdj(seu, platform = 'ST')
if(class(seu@assays$RNA)=="Assay5"){
  var.features <- seu@assays$RNA@meta.data$var.features
  var.features <- var.features[!is.na(var.features)]
  dat <- GetAssayData(seu, assay = "RNA", slot='data')
  X <- Matrix::t(dat[var.features,])
}else{
  var.features <- seu@assays$RNA@var.features
  X <- Matrix::t(seu[["RNA"]@data[var.features,])
}

# maxIter = 2 is only used for illustration, and user can use default.
drscList <- DR.SC_fit(X,Adj_sp=Adj_sp, K=4, maxIter=2, verbose=TRUE)
```

drscPlot

*tNSE or UMAP plot visualization***Description**

Intuitive way of visualizing how cell types changes across the embeddings obatined by DR-SC.

Usage

```
drscPlot(seu, dims=1:5, visu.method='tSNE',...)
```

Arguments

seu	an object of class "Seurat" obtained by DR.SC .
dims	a positive integer to specify the number of latent features for visualiztion.
visu.method	a string including 'tSNE' or "UMAP".
...	Other arguments passing to DimPlot function.

Details

Nothing

Value

return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
data(seu)
library(Seurat)
seu <- NormalizeData(seu)
# choose spatially variable features
seu <- FindSVGs(seu)

# use SVGs to fit DR.SC model
# maxIter = 2 is only used for illustration, and user can use default.
seu1 <- DR.SC(seu, K=4,platform = 'ST', maxIter = 2,verbose=FALSE)
drscPlot(seu1)
```

FindSVGs

Find spatially variable genes

Description

Identifies features that have spatially variation along spots using SPARK-X.

Usage

```
FindSVGs(seu, nfeatures=2000, covariates=NULL,
          preHVGs=5000,num_core=1, verbose=TRUE)
```


Arguments

seu	an object of class "Seurat".
nfeatures	a positive integer, means how many spatially variable genes to be chosen. If there are less than 2000 features in seu, then all features are identified.
covariates	a covariate matrix named control variable matrix whose number of rows is equal to the number of columns of seu.
preHVGs	a positive integer, the number of highly variable genes selected for speeding up computation of SPARK-X in selecting spatially variable features.
num_core	an optional positive integer, specify the cores used for identifying the SVGs in parallel.
verbose	an optional logical value, whether output the related information.

Details

Nothing

Value

return a revised Seurat object by adding three columns named "is.SVGs", "order.SVGs" and "adjusted.pval.SVGs" in the meta.features of default Assay.

Note

nothing

References

Zhu, J., Sun, S., Zhou, X.: Spark-x: non-parametric modeling enables scalable and robust detection of spatialexpression patterns for large spatial transcriptomic studies. *Genome Biology* 22(1), 1-25 (2021)

See Also

[topSVGs](#)

Examples

```
data(seu)
seu<-FindSVGs(seu, nfeatures=100)
topSVGs(seu)
```

getAdj	<i>Calculate the adjacency matrix given the spatial coordinates</i>
--------	---

Description

Calculate the adjacency matrix for the spatial transcriptomics data measured on 10X Visium, ST or other platforms as a Seurat object.

Usage

```
## S3 method for class 'Seurat'
getAdj(obj, platform = c('Visium', "ST", "Other_SRT"), ...)
```

Arguments

obj	an object with class "Seurat", there are spatial coordinates information in the metadata of obj, named "row" and "col", where first column is x-axis coordinate, the second column is y-axis coordinate. getAdj_manual and getAdj_auto supports multi-dimensional spatial coordinates with a matrix as input.
platform	a string, specify the platform of the provided data, default as "Visium". There are more platforms to be chosen, including ("Visium", "ST", "Other_SRT"), which means there are spatial coordinates information in the metadata of obj, named "row" and "col". The platform helps to calculate the adjacency matrix by defining the neighborhoods.
...	Other arguments to pass into getAdj_auto function.

Details

For lattice grids in ST platform (ST), the interior spot has four neighbors (left, right, up and down), the boundary spot has three neighbors, and the spot in the corner has two neighbors. For hexagon grids, such as spatial coordinate in 10X Visium platform (Visium), the interior spot has six neighbors. For the irregular coordinates in other platforms (Other_SRT), Euclidean distance is adopted to decide whether a spot is a neighbor of another spot. For example, if the Euclidean distance between spot A and B is less than a radius, then A is taken as the neighbourhood of B. See functions [getAdj_auto](#) and [getAdj_manual](#) for more details.

Value

Return a `dgCMatrix` object recording the information of neighborhoods about each spot.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Xu Liao, Yi Yang, Huazhen Lin, Joe Yeong, Xiang Zhou, Xingjie Shi & Jin Liu (2022). Joint dimension reduction and clustering analysis of single-cell RNA-seq and spatial transcriptomics data, *Nucleic Acids Research*, gkac219.

See Also

[getAdj_auto](#), [getAdj_manual](#).

Examples

```
## S3 method for class "Seurat"
data(seu)
Adj_sp <- getAdj(seu, platform = 'ST')
```

getAdj_auto

Calculate adjacency matrix by automatically choosing radius

Description

an efficient function to find the radius by bi-section method , then find neighbors based on the matrix of position, which ensures that each spot has approximately lower.med~upper.med neighbors in the sense of median.

Usage

```
getAdj_auto(pos, lower.med=4, upper.med=6, radius.upper= NULL)
```

Arguments

pos	a n-by-2 matrix of position.
lower.med	an integer, the lower bound of median number of neighbors among all spots.
upper.med	an integer, the upper bound of median number of neighbors among all spots.
radius.upper	a real, the upper bound of radius, default as NULL. If radius.upper= NULL, the upper bound is automatically determined by algorithm.

Value

A sparse adjacency matrix containing the neighbourhood.

See Also

[getAdj](#), [getAdj_manual](#).

getAdj_manual	<i>Calculate adjacency matrix by user-specified radius</i>
---------------	--

Description

an efficient function to find the neighbors based on the matrix of position and a pre-defined radius.

Usage

```
getAdj_manual(pos, radius)
```

Arguments

pos	is a n-by-d matrix of position, where n is the number of spots, and d is the dimension of coordinates.
radius	is a threshold of Euclidean distance to decide whether a spot is an neighborhood of another spot. For example, if the Euclidean distance between spot A and B is less than radius, then A is taken as the neighbourhood of B.

Value

A sparse matrix containing the neighbourhood

See Also

[getAdj_auto](#), [getAdj](#).

getneighborhood_fast	<i>getneighborhood_fast</i>
----------------------	-----------------------------

Description

an efficient function to find the neighborhood based on the matrix of position and a pre-defined cutoff

Usage

```
getneighborhood_fast(x, radius)
```

Arguments

x	is a n-by-2 matrix of position.
radius	is a threshold of Euclidean distance to decide whether a spot is an neighborhood of another spot. For example, if the Euclidean distance between spot A and B is less than cutoff, then A is taken as the neighbourhood of B.

Value

A sparse matrix containing the neighbourhood

mbicPlot	<i>MBIC plot visualization</i>
----------	--------------------------------

Description

Intuitive way of visualizing how modified BIC values changes across different number of clusters

Usage

```
mbicPlot(seu, criteria="MBIC")
```

Arguments

seu	an object of class Seurat revised by DR.SC with argument K=NULL.
criteria	a string specifying the information criteria such as AIC, BIC and MBIC to be plotted, default as MBIC.

Details

Nothing

Value

return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
library(Seurat)
data(seu)
seu <- NormalizeData(seu)
# choose spatially variable features
seu <- FindSVGs(seu)
## Just for illustrating the usage of mbicPlot
seu[["RNA"]][misc[['icMat']] <- data.frame(K=2:5, MBIC=c(105, 101, 99, 108))
mbicPlot(seu)
```

read10XVisium	<i>Read the spatial transcriptomics data measured on 10X Visium platform</i>
---------------	--

Description

Read the spatial transcriptomics data measured on 10X Visium platform as a Seurat object, where the spatial coordinates are saved in the metadata, named "row" and "col".

Usage

```
read10XVisium(dirname)
```

Arguments

dirname	A string, the dictory of Visium datasets
---------	--

Details

Nothing

Value

return a Seurat object.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

None

Examples

```
## Not run:
## set your file directory, then read it.
data_name <- "D/HCC"
HCC1 <- read10XVisium(data_name)

## End(Not run)
```

readscRNAseq

Read the scRNAseq data measured on scRNA sequencing platform

Description

Read the single cell RNA sequencing data measured on scRNA sequencing platform as a Seurat object.

Usage

```
readscRNAseq(mtx, cells, features, ...)
```

Arguments

mtx	a string, ame or remote URL of the mtx file
cells	a string, Name or remote URL of the cells/barcodes file
features	a string, Name or remote URL of the features/genes file
...	the arguments passing to ReadMtx

Details

Nothing

Value

return a Seurat object including expression matrix.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

None

Examples

```
## Not run:
### set the file directory, then read it.
seu <- readscRNAseq(mtx="GSM3755564_16_Liver_Treg_matrix.mtx.gz",
                    features='GSM3755564_16_Liver_Treg_genes.tsv.gz',
                    cells='GSM3755564_16_Liver_Treg_barcodes.tsv.gz' )

seu

## End(Not run)
```

RunWPCA

*Run Weighted Principal Component Analysis***Description**

Run a weighted PCA dimensionality reduction

Usage

```
RunWPCA(object, q=15)
### S3 method for class "Seurat"
## RunWPCA(object, q=15)

### S3 method for class "matrix"
## RunWPCA(object, q=15)

### S3 method for class "dgCMatrx"
## RunWPCA(object, q=15)
```

Arguments

object	an object named "Seurat", "matrix" or "dgCMatrx". The object of class "Seurat" must include slot "scale.data".
q	an optional positive integer, specify the number of features to be extracted.

Details

Nothing

Value

For Seurat object, return a Seurat object. For object "matrix" and "dgCMatrix", return a object "matrix" with rownames same as the colnames of X, and colnames "WPCA1" to "WPCAq".

Note

nothing

Author(s)

Wei Liu

References

Bai, J. and Liao, Y. (2017). Inferences in panel data with interactive effects using large covariance matrices. *Journal of Econometrics*, 200(1):59–78.

See Also

None

Examples

```
## Not run:
library(Seurat)
data(seu)
## log-normalization
seu <- NormalizeData(seu)
##
seu <- FindVariableFeatures(seu, nfeatures=80)
## Scale
seu <- ScaleData(seu)
## Run WPCA
seu <- RunWPCA(seu)
seu
## Run tSNE based on wpca
seu <- RunTSNE(seu, reduction='wpca')
seu
## Find SVGs
seu <- FindSVGs(seu, nfeatures=80)
(genes <- topSVGs(seu, ntop=10))
Idents(seu) <- factor(paste0("cluster", seu$true_clusters), levels=paste0("cluster",1:4))
RidgePlot(seu, features = genes[1:2], ncol = 2)
FeaturePlot(seu, features = genes[1:2], reduction = 'tsne' ,ncol=2)

## End(Not run)
```

selectModel	<i>Select the number of clusters</i>
-------------	--------------------------------------

Description

Select the number of clusters by specified criteria.

Usage

```
selectModel(obj, criteria = 'MBIC', pen.const=1)
## S3 method for class 'drscObject'
selectModel(obj, criteria = 'MBIC', pen.const=1)
## S3 method for class 'Seurat'
selectModel(obj, criteria = 'MBIC', pen.const=1)
```

Arguments

S	
obj	an object with class Seurat by DR.SC or class drscObject by DR.SC_fit .
criteria	a string, specify the criteria used for selecting the number of clusters, supporting "MBIC", "BIC" and "AIC".
pen.const	an optional positive value, the adjusted constant used in the MBIC criteria. It usually takes value between 0.1 to 1.

Value

For S3 method of Seurat, it return a revised "Seurat" object with updated Idents(seu), spatial.drsc.cluster in the metadata and DimReduc object named dr-sc in the slot reductions. For S3 method of drscObject, it returns a list with the following components:

bestK	the selected number of clusters.
cluster	inferred class labels
hZ	extracted latent features.
icMat	a numeric matrix including the criteria value for each number of clusters K.

Note

nothing

Author(s)

Wei Liu

References

Wei Liu, Xu Liao, Yi Yang, Huazhen Lin, Joe Yeong, Xiang Zhou, Xingjie Shi & Jin Liu (2022). Joint dimension reduction and clustering analysis of single-cell RNA-seq and spatial transcriptomics data, *Nucleic Acids Research*, gkac219.

See Also

[DR.SC](#), [DR.SC_fit](#).

Examples

```
data(seu)
library(Seurat)
seu <- NormalizeData(seu, verbose=FALSE)
# or choose 40 spatailly variable features using FindSVGs in DR.SC
seu <- FindSVGs(seu, nfeatures = 40, verbose=FALSE)
# users define the adjacency matrix
Adj_sp <- getAdj(seu, platform = 'ST')
dat <- GetAssayData(seu, assay = "RNA", slot='data')
X <- Matrix::t(dat)
# maxIter = 2 is only used for illustration, and user can use default.
drscList <- DR.SC_fit(X, Adj_sp=Adj_sp, K=4, maxIter=2, verbose=TRUE)
drsc1 <- selectModel(drscList)
str(drsc1)
```

seu

A simulated spatial transcriptomics data

Description

This data is a simulated spatial transcriptomics data based on ST platform, has 100 spots and 50 features.

Note

nothing

Author(s)

Wei Liu

References

None

Examples

```
data(seu)
```

spatialPlotClusters	<i>Spatial coordinates plot visualization</i>
---------------------	---

Description

Intuitive way of visualizing how cell types changes across the spatial locations.

Usage

```
spatialPlotClusters(seu)
```

Arguments

seu	an object of class "Seurat" obtained by DR.SC .
-----	---

Details

Nothing

Value

return a ggplot2 object.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

None

Examples

```
## we generate the spatial transcriptomics data with lattice neighborhood, i.e. ST platform.
data(seu)
library(Seurat)
seu <- NormalizeData(seu)
# choose spatially variable features using Seurat
seu <- FindSVGs(seu)
# use SVGs to fit DR.SC model
# maxIter = 2 is only used for illustration, and user can use default.
```

```
seu1 <- DR.SC(seu, K=4,platform = 'ST', maxIter=2,verbose=FALSE)
spatialPlotClusters(seu1)
```

sp_means_Rcpp*Calculate column-wise or row-wise mean*

Description

Calculate column-wise or row-wise mean

Usage

```
sp_means_Rcpp(sp_data, rowMeans = FALSE)
```

Arguments

sp_data	A sparse matrix
rowMeans	A boolean value, whether to calculate row-wise mean

Value

A $n \times 1$ or $p \times 1$ matrix

sp_sums_Rcpp*Calculate column-wise or row-wise sum*

Description

Calculate column-wise or row-wise sum

Usage

```
sp_sums_Rcpp(sp_data, rowSums = FALSE)
```

Arguments

sp_data	A sparse matrix
rowSums	A boolean value, whether to calculate row-wise sum

Value

A $n \times 1$ or $p \times 1$ matrix

topSVGs	<i>Return the top n SVGs</i>
---------	------------------------------

Description

Return top n spatially variable genes given a Seurat object performed by [FindSVGs](#).

Usage

```
topSVGs(seu, ntop=5)
```

Arguments

seu	an object of class "Seurat".
ntop	an optional positive integer, means how many spatially variable genes to access.

Details

Nothing

Value

return a [character](#) vector including the names of SVGs.

Note

nothing

Author(s)

Wei Liu

References

None

See Also

[topSVGs](#)

Examples

```
data(seu)
seu <- FindSVGs(seu, nfeatures=100, verbose=FALSE)
(genes <- topSVGs(seu, ntop=10))
```

Index

- * **Adjacency matrix**
 - getAdj, [10](#)
 - * **DR-SC**
 - drscPlot, [7](#)
 - * **Dimension reduction**
 - DR.SC, [3](#)
 - DR.SC_fit, [5](#)
 - RunWPCA, [16](#)
 - * **Hexagon grids**
 - getAdj, [10](#)
 - * **Input of scRNAseq data**
 - readscRNAseq, [15](#)
 - * **Input of spatial data**
 - read10XVisium, [14](#)
 - * **Lattice grids**
 - getAdj, [10](#)
 - * **Modified BIC**
 - mbicPlot, [13](#)
 - * **SVGs**
 - topSVGs, [22](#)
 - * **Spatial clustering**
 - DR.SC, [3](#)
 - * **Spatial plot**
 - spatialPlotClusters, [20](#)
 - * **UMAP**
 - drscPlot, [7](#)
 - * **Weighted PCA**
 - RunWPCA, [16](#)
 - * **spatial clustering and cell type identification**
 - DR.SC_fit, [5](#)
 - * **tSNE**
 - drscPlot, [7](#)
- character, [22](#)
- DimPlot, [7](#)
- dlpfc151510, [2](#)
- DR.SC, [3](#), [7](#), [13](#), [18–20](#)
- DR.SC_fit, [3](#), [5](#), [18](#), [19](#)
- drscPlot, [7](#)
- FindSVGs, [8](#), [22](#)
- getAdj, [4](#), [10](#), [11](#), [12](#)
- getAdj_auto, [10](#), [11](#), [11](#), [12](#)
- getAdj_manual, [10](#), [11](#), [12](#)
- getneighborhood_fast, [12](#)
- list, [6](#)
- mbicPlot, [13](#)
- read10XVisium, [14](#)
- ReadMtx, [15](#)
- readscRNAseq, [15](#)
- RunWPCA, [16](#)
- selectModel, [3](#), [18](#)
- seu, [19](#)
- sp_means_Rcpp, [21](#)
- sp_sums_Rcpp, [21](#)
- spatialPlotClusters, [20](#)
- topSVGs, [9](#), [22](#), [22](#)