

# Package ‘RegionalST’

May 17, 2024

**Type** Package

**Title** Investigating regions of interest and performing cross-regional analysis with spatial transcriptomics data

**Version** 1.2.0

**Description** This package analyze spatial transcriptomics data through cross-regional analysis. It selects regions of interest (ROIs) and identifies cross-regional cell type-specific differential signals. The ROIs can be selected using automatic algorithm or through manual selection. It facilitates manual selection of ROIs using a shiny application.

**License** GPL-3

**Encoding** UTF-8

**LazyData** FALSE

**RoxygenNote** 7.2.3

**biocViews** Spatial, Transcriptomics, Reactome, KEGG

**Depends** R (>= 4.3.0)

**Imports** stats, grDevices, utils, ggplot2, dplyr, scater, gridExtra, BayesSpace, fgsea, magrittr, SingleCellExperiment, RColorBrewer, Seurat, S4Vectors, tibble, TOAST, assertthat, colorspace, shiny, SummarizedExperiment

**Suggests** BiocStyle, knitr, rmarkdown, gplots, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/RegionalST>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 4a6fc9d

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-16

**Author** Ziyi Li [aut, cre]

**Maintainer** Ziyi Li <zli16@mdanderson.org>

## Contents

DoGSEA	2
DrawDotplot	3
DrawRegionProportion	4
DrawRegionProportion_withProp	4
exampleRes	5
example_sce	6
FindRegionalCells	6
GetCrossRegionalDE_raw	7
GetCrossRegionalDE_withProp	8
GetOneRadiusEntropy	9
GetOneRadiusEntropy_withProp	11
getProportion	12
ManualSelectCenter	12
mySpatialPreprocess	13
pathways_hallmark	14
pathways_kegg	14
pathways_reactome	15
PlotOneSelectedCenter	16
RankCenterByEntropy	16
RankCenterByEntropy_withProp	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

DoGSEA	<i>Perform GSEA analysis for cross-regional DE genes</i>
--------	--

---

### Description

Perform GSEA analysis for cross-regional DE genes

### Usage

```
DoGSEA(considerRes, whichDB = "hallmark", gmdir = NULL, withProp = FALSE)
```

### Arguments

considerRes	A list of cross-regional DE genes.
whichDB	A character string to select the database names, e.g., "hallmark", "kegg", "reactome".
gmdir	Directory for external database gmt file location.
withProp	Whether deconvolution proportion is used in previous steps.

### Value

A list including GSEA results for all cell types.

**Examples**

```
data(exampleRes)
allCTres <- DoGSEA(exampleRes, whichDB = "hallmark", withProp = TRUE)
```

---

**DrawDotplot***Draw dot plot for GSEA results of cross-regional DE genes*

---

**Description**

Draw dot plot for GSEA results of cross-regional DE genes

**Usage**

```
DrawDotplot(  
  allCTres,  
  CT = 1,  
  angle = 20,  
  vjust = 0.9,  
  hjust = 1,  
  padj_cutoff = 1,  
  topN = 20,  
  chooseP = "padj",  
  eachN = NULL  
)
```

**Arguments**

<code>allCTres</code>	A list of GSEA results for all cell types.
<code>CT</code>	A number of the interested cell type, e.g., 1, 2, 3.
<code>angle</code>	A number of plotting parameter, angle of the x axis label.
<code>vjust</code>	A number of vertical adjustment in plotting.
<code>hjust</code>	A number of horizontal adjustment in plotting.
<code>padj_cutoff</code>	A cutoff number of adjusted p value.
<code>topN</code>	A number of the plotted top pathways.
<code>chooseP</code>	A character string for the p value that used in plotting, e.g., "padj" or "pval".
<code>eachN</code>	The maximum number of pathways in each cell type.

**Value**

A plot object

**Examples**

```
data(exampleRes)
allCTres <- DoGSEA(exampleRes, whichDB = "hallmark", withProp = TRUE)
DrawDotplot(allCTres, CT = 1, angle = 15, vjust = 1, chooseP = "padj")
```

---

DrawRegionProportion *Draw regional cell type distribution with cell type annotation information*

---

**Description**

Draw regional cell type distribution with cell type annotation information

**Usage**

```
DrawRegionProportion(sce, label = "celltype", selCenter = seq_len(10))
```

**Arguments**

sce            A single cell experiment object.  
 label         A string character for the cell type variable.  
 selCenter     A vector of the interested ROIs, e.g., 1:4.

**Value**

A plot object.

**Examples**

```
data("example_sce")
DrawRegionProportion(example_sce, label = "celltype", selCenter = 1:3)
```

---

DrawRegionProportion\_withProp  
*Draw regional cell type distribution with cellular proportion information*

---

**Description**

Draw regional cell type distribution with cellular proportion information

**Usage**

```
DrawRegionProportion_withProp(  
  sce,  
  label = "CARD_CellType",  
  selCenter = seq_len(10)  
)
```

**Arguments**

sce	A single cell experiment object.
label	A string character for the cell type variable.
selCenter	A vector of the interested ROIs, e.g., 1:4.

**Value**

A plot object.

**Examples**

```
data("example_sce")  
DrawRegionProportion_withProp(example_sce,  
                               label = "Proportions",  
                               selCenter = 1:3)
```

---

exampleRes

*Example DE output*

---

**Description**

A simulated example DE output file

**Usage**

```
data(exampleRes)
```

**Format**

A list object.

**Value**

A list object.

**Examples**

```
data(exampleRes)
```

---

`example_sce`*Example single cell experiment for input*

---

**Description**

A simulated example input data file

**Usage**

```
data(example_sce)
```

**Format**

A SingleCellExperiment object.

**Value**

A SingleCellExperiment object.

**Examples**

```
data(example_sce)
```

---

`FindRegionalCells`*Identify regional cells given centers and radiuses*

---

**Description**

Identify regional cells given centers and radiuses

**Usage**

```
FindRegionalCells(  
  sce,  
  centerID,  
  enhanced = FALSE,  
  radius = 10,  
  avern = 5,  
  doPlot = FALSE,  
  returnPlot = FALSE  
)
```

**Arguments**

sce	A single cell experiment object.
centerID	One or a vector of spot IDs as centers of ROIs.
enhanced	A logical variable for plotting enhanced plot or now. Default is FALSE.
radius	A number of fixed ROI radius.
avern	A number of the average sites used to compute unit distance, default is 5.
doPlot	A logical variable to specify whether plot the figure or not.
returnPlot	a logical variable to specify whether output the plot or not.

**Value**

A list including center spot ID and regional spot IDs.

**Examples**

```
# FindRegionalCells(sce, centerID = "ACGCCTGACACGCGCT-1")
```

---

GetCrossRegionalDE\_raw

*Identify cross-regional differential analysis*

---

**Description**

Identify cross-regional differential analysis

**Usage**

```
GetCrossRegionalDE_raw(  
  sce,  
  twoCenter = c(3, 4),  
  enhanced = FALSE,  
  label = "celltype",  
  n_markers = 10,  
  logfc.threshold = 0.25,  
  angle = 30,  
  hjust = 0,  
  size = 3,  
  min.pct = 0.1,  
  padj_filter = 0.05,  
  doHeatmap = TRUE  
)
```

**Arguments**

sce	A single cell experiment object.
twoCenter	A vector of two numbers for the interested ROI numbers.
enhanced	A logical variable for using enhanced data or not.
label	A variable name that contains the cell type information.
n_markers	A number specifying the top DE gene number.
logfc.threshold	A number for the cutoff threshold of log fold change.
angle	A number for angle when plotting.
hjust	A number for horizontal justification when plotting.
size	A number for text font size.
min.pct	A number of minimum percentage specified in the Seurat DE function.
padj_filter	A number for filtering adjusted p values.
doHeatmap	Logical variable for whether drawing the heatmap.

**Value**

A list including the top DE genes (topDE), and all DE genes (allDE).

**Examples**

```
data("example_sce")
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
# I used a very big padj filter here because this is just a toy data
GetCrossRegionalDE_raw(example_sce, twoCenter = c(1,2),
  min.pct = 0.01, logfc.threshold = 0.01,
  padj_filter = 0.5)
```

---

GetCrossRegionalDE\_withProp

*Identify cross-regional differential analysis with proportion*

---

**Description**

Identify cross-regional differential analysis with proportion



**Usage**

```
GetCrossRegionalDE_withProp(
  sce,
  twoCenter = c(3, 4),
  label = "celltype",
  n_markers = 10,
  angle = 30,
  hjust = 0,
  size = 3,
  padj_filter = 0.05,
  doHeatmap = TRUE
)
```

**Arguments**

sce	A single cell experiment object.
twoCenter	A vector of two numbers for the interested ROI numbers.
label	A variable name that contains the cell type information.
n_markers	A number specifying the top DE gene number.
angle	A number for angle when plotting.
hjust	A number for horizontal justification when plotting.
size	A number for text font size.
padj_filter	A number for filtering adjusted p values.
doHeatmap	Logical variable for whether drawing the heatmap.

**Value**

A list including the top DE genes (topDE), and all DE genes (allDE).

**Examples**

```
data("example_sce")
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
# Since the example data is very small, I set padj filter as NULL. Default is 0.05.
GetCrossRegionalDE_withProp(example_sce, twoCenter = c(1,2), padj_filter = NULL)
```

---

GetOneRadiusEntropy     *Computer the entropy for a fixed radius*

---

**Description**

Computer the entropy for a fixed radius

**Usage**

```

GetOneRadiusEntropy(
  sce,
  selectN,
  enhanced = FALSE,
  weight = NULL,
  label = "celltype",
  radius = 10,
  doPlot = FALSE,
  mytitle = NULL
)

```

**Arguments**

sce	A single cell experiment object.
selectN	A total number for selected centers. Should be smaller than the total site number.
enhanced	A logical variable of whether using enhanced data.
weight	A data frame to specify the weights of all cell types.
label	A variable name that contains the cell type information.
radius	A number for fixed radius.
doPlot	Logical variable about whether draw the plot.
mytitle	A character string for the title of the plot.

**Value**

A list including the selected centers, computed entropies, radius.

**Examples**

```

data("example_sce")
weight <- data.frame(celltype = c("Cancer Epithelial", "CAFs",
                                "T-cells", "Endothelial",
                                "PVL", "Myeloid", "B-cells",
                                "Normal Epithelial", "Plasmablasts"),
                    weight = c(0.25,0.05,
                               0.25,0.05,
                               0.025,0.05,
                               0.25,0.05,0.025))
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
GetOneRadiusEntropy(example_sce, selectN = round(length(example_sce$spot)/2),
                    weight = weight, radius = 5, doPlot = TRUE,
                    mytitle = "Radius 5 weighted entropy")

```

---

 GetOneRadiusEntropy\_withProp

*Computer the entropy for a fixed radius with cell type proportion*


---

## Description

Computer the entropy for a fixed radius with cell type proportion

## Usage

```
GetOneRadiusEntropy_withProp(
  sce,
  selectN,
  weight = NULL,
  label = "celltype",
  radius = 10,
  doPlot = FALSE,
  mytitle = NULL
)
```

## Arguments

sce	A single cell experiment object.
selectN	A total number for selected centers. Should be smaller than the total site number.
weight	A data frame to specify the weights of all cell types.
label	A variable name that contains the cell type information.
radius	A number for fixed radius.
doPlot	Logical variable about whether draw the plot.
mytitle	A character string for the title of the plot.

## Value

A list including the selected centers, computed entropies, radius.

## Examples

```
data("example_sce")
weight <- data.frame(celltype = c("Cancer Epithelial", "CAFs", "T-cells", "Endothelial",
  "PVL", "Myeloid", "B-cells", "Normal Epithelial", "Plasmablasts"),
  weight = c(0.25,0.05,
    0.25,0.05,
    0.025,0.05,
    0.25,0.05,0.025))
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
GetOneRadiusEntropy_withProp(example_sce, selectN = round(length(example_sce$spot)/10),
  weight = weight,
```

```
radius = 5,
doPlot = TRUE,
mytitle = "Radius 5 weighted entropy")
```

---

getProportion      *Define an accessor method for Proportion\_CARD*

---

### Description

Define an accessor method for Proportion\_CARD

### Usage

```
getProportion(card)
```

### Arguments

card              A CARD object.

### Value

A matrix containing the spot-level cell type proportion information

### Examples

```
# getProportion(card)
```

---

ManualSelectCenter      *Manually select top ROIs*

---

### Description

Manually select top ROIs

### Usage

```
ManualSelectCenter(sce)
```

### Arguments

sce              A single cell experiment object.

### Value

An sce object with selected centers and radiuses.

**Examples**

```
data("example_sce")
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
# I commented this out because the shiny app will get stuck without input.
# example_sce <- ManualSelectCenter(example_sce)
```

---

mySpatialPreprocess    *Perform Preprocessing for spatial data (tailored from BayesSpace function)*

---

**Description**

Perform Preprocessing for spatial data (tailored from BayesSpace function)

**Usage**

```
mySpatialPreprocess(
  sce,
  platform = c("Visium", "ST"),
  n.PCs = 15,
  n.HVGs = 2000,
  skip.PCA = FALSE,
  assay.type = "logcounts"
)
```

**Arguments**

sce	A SingleCellExperiment object.
platform	Which platform the data are from, Visium or ST.
n.PCs	Number of PCs used in the analysis.
n.HVGs	Number of highly variable genes used in the analysis.
skip.PCA	A boolean variable to choose whether skipping the PCA step or not.
assay.type	Which assay to use, default is logcounts.

**Value**

A processed SingleCellExperiment object.

**Examples**

```
data(example_sce)
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
```

---

pathways\_hallmark      *Hallmark database*

---

**Description**

Hallmark database downloaded from MSigDB (Feb, 2023)

**Usage**

```
data(pathways_hallmark)
```

**Format**

A list object.

**Value**

A list object.

**Source**

**MSigDB**

**References**

Liberzon et al. (2015) Cell Syst. 1(6):417-425 ([PubMed](#))

**Examples**

```
data(pathways_hallmark)
```

---

pathways\_kegg      *KEGG database*

---

**Description**

KEGG database downloaded from MSigDB (Feb, 2023)

**Usage**

```
data(pathways_kegg)
```

**Format**

A list object.

**Value**

A list object.

**Source**

[MSigDB](#)

**References**

Kanehisa and Goto (2000) Nucleic Acids Research 28(1):27-30 ([PubMed](#))

**Examples**

```
data(pathways_kegg)
```

---

pathways_reactome	<i>REACTOME database</i>
-------------------	--------------------------

---

**Description**

REACTOME database downloaded from MSigDB (Feb, 2023)

**Usage**

```
data(pathways_reactome)
```

**Format**

A list object.

**Value**

A list object.

**Source**

[MSigDB](#)

**References**

Jassal et al. (2020) Nucleic Acids Research 28(1):27-30 ([PubMed](#))

**Examples**

```
data(pathways_reactome)
```

---

PlotOneSelectedCenter *Plot one selected ROI*

---

### Description

Plot one selected ROI

### Usage

```
PlotOneSelectedCenter(sce, ploti, enhanced = FALSE)
```

### Arguments

sce	A single cell experiment object.
ploti	A number of indicate which ROI to plot.
enhanced	A logical variable for using enhanced data or not.

### Value

A figure object for the selected ROI.

### Examples

```
data("example_sce")
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
PlotOneSelectedCenter(example_sce, ploti = 1)
```

---

RankCenterByEntropy *Automatically rank ROI centers based on entropy*

---

### Description

Automatically rank ROI centers based on entropy

### Usage

```
RankCenterByEntropy(
  sce,
  weight,
  enhanced = FALSE,
  selectN = round(length(sce$spot)/10),
  label = "celltype",
  topN = 10,
  min_radius = 10,
  avern = 5,
```



```

    radius_vec = c(10, 15, 20),
    doPlot = TRUE
  )

```

### Arguments

sce	A single cell experiment object.
weight	A data frame to specify the weights of all cell types.
enhanced	A logical variable of whether using enhanced data.
selectN	A total number for selected centers. Should be smaller than the total site number.
label	A variable name that contains the cell type information.
topN	A number to specify the total amount of top ranked ROIs.
min_radius	The minimum repellent radius.
avern	A number of the average sites used to compute unit distance, default is 5.
radius_vec	A vector of numbers for candidate radiuses.
doPlot	Logical variable about whether draw the plot.

### Value

An sce object with selected ROI information.

### Examples

```

data("example_sce")
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
weight <- data.frame(celltype = c("Cancer Epithelial", "CAFs", "T-cells", "Endothelial",
  "PVL", "Myeloid", "B-cells", "Normal Epithelial", "Plasmablasts"),
  weight = c(0.25,0.05,
    0.25,0.05,
    0.025,0.05,
    0.25,0.05,0.025))
example_sce <- RankCenterByEntropy(example_sce, weight, label = "celltype",
  selectN = round(length(example_sce$spot)/10),
  topN = 3, min_radius = 10,
  radius_vec = c(10,15),
  doPlot = TRUE)

```

---

RankCenterByEntropy\_withProp

*Automatically rank ROI centers based on entropy with proportions*

---

### Description

Automatically rank ROI centers based on entropy with proportions

**Usage**

```
RankCenterByEntropy_withProp(
  sce,
  weight,
  selectN = round(length(sce$spot)/10),
  topN = 10,
  min_radius = 10,
  avern = 5,
  radius_vec = c(10, 15, 20),
  doPlot = TRUE
)
```

**Arguments**

sce	A single cell experiment object.
weight	A data frame to specify the weights of all cell types.
selectN	A total number for selected centers. Should be smaller than the total site number.
topN	A number to specify the total amount of top ranked ROIs.
min_radius	The minimum repellent radius.
avern	A number of the average sites used to compute unit distance, default is 5.
radius_vec	A vector of numbers for candidate radiuses.
doPlot	Logical variable about whether draw the plot.

**Value**

An sce object with selected ROI information.

**Examples**

```
data("example_sce")
weight <- data.frame(celltype = c("Cancer Epithelial", "CAFs", "T-cells", "Endothelial",
  "PVL", "Myeloid", "B-cells", "Normal Epithelial", "Plasmablasts"),
  weight = c(0.25,0.05,
    0.25,0.05,
    0.025,0.05,
    0.25,0.05,0.025))
example_sce <- mySpatialPreprocess(example_sce, platform="Visium")
## I set our min_raius as 10 and radius vector as 10 and 15 as the example dataset is very small
example_sce <- RankCenterByEntropy_withProp(example_sce, weight,
  selectN = round(length(example_sce$spot)/10),
  topN = 3, min_radius = 10,
  radius_vec = c(10,15),
  doPlot = TRUE)
```

# Index

## \* datasets

- example\_sce, [6](#)
- exampleRes, [5](#)
- pathways\_hallmark, [14](#)
- pathways\_kegg, [14](#)
- pathways\_reactome, [15](#)

- DoGSEA, [2](#)
- DrawDotplot, [3](#)
- DrawRegionProportion, [4](#)
- DrawRegionProportion\_withProp, [4](#)

- example\_sce, [6](#)
- exampleRes, [5](#)

- FindRegionalCells, [6](#)

- GetCrossRegionalDE\_raw, [7](#)
- GetCrossRegionalDE\_withProp, [8](#)
- GetOneRadiusEntropy, [9](#)
- GetOneRadiusEntropy\_withProp, [11](#)
- getProportion, [12](#)

- ManualSelectCenter, [12](#)
- mySpatialPreprocess, [13](#)

- pathways\_hallmark, [14](#)
- pathways\_kegg, [14](#)
- pathways\_reactome, [15](#)
- PlotOneSelectedCenter, [16](#)

- RankCenterByEntropy, [16](#)
- RankCenterByEntropy\_withProp, [17](#)