

# Package ‘scider’

November 6, 2025

**Type** Package

**Title** Spatial cell-type inter-correlation by density in R

**Version** 1.9.0

**Description** scider is an user-friendly R package providing functions to model the global density of cells in a slide of spatial transcriptomics data. All functions in the package are built based on the SpatialExperiment object, allowing integration into various spatial transcriptomics-related packages from Bioconductor. After modelling density, the package allows for serveral downstream analysis, including colocalization analysis, boundary detection analysis and differential density analysis.

**biocViews** Spatial, Transcriptomics

**License** GPL-3 + file LICENSE

**URL** <https://github.com/ChenLaboratory/scider>,  
<https://chenlaboratory.github.io/scider/>

**BugReports** <https://github.com/ChenLaboratory/scider/issues>

**Encoding** UTF-8

**LazyData** false

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Imports** SpatialExperiment, SummarizedExperiment, spatstat.geom, spatstat.explore, sf, lwgeom, SpatialPack, ggplot2, stats, pheatmap, plotly, shiny, igraph, janitor, knitr, methods, utils, isoband, S4Vectors, grDevices, dbscan, hexDensity, hexbin, uwot, SingleCellExperiment, BiocNeighbors, irlba

**Suggests** edgeR, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 4.3)

**VignetteBuilder** knitr

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

**git\_branch** devel

**git\_last\_commit** e0a2a6b

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2025-11-05

**Author** Mengbo Li [aut] (ORCID: <<https://orcid.org/0000-0002-9666-5810>>),

Ning Liu [aut] (ORCID: <<https://orcid.org/0000-0002-9487-9305>>),

Quoc Hoang Nguyen [aut] (ORCID:

<<https://orcid.org/0009-0007-2828-7567>>),

Yunshun Chen [aut, cre] (ORCID:

<<https://orcid.org/0000-0003-4911-5653>>)

**Maintainer** Yunshun Chen <yuchen@wehi.edu.au>

## Contents

|                             |    |
|-----------------------------|----|
| scider-package . . . . .    | 3  |
| allocateCells . . . . .     | 4  |
| cellsInRegion . . . . .     | 5  |
| computeDensity . . . . .    | 5  |
| computeDensityHex . . . . . | 6  |
| contour2sf . . . . .        | 7  |
| coord_hash . . . . .        | 8  |
| corDensity . . . . .        | 8  |
| findNbrsGrid . . . . .      | 9  |
| findNbrsSNN . . . . .       | 10 |
| findNbrsSpatial . . . . .   | 12 |
| findROI . . . . .           | 13 |
| getClusters . . . . .       | 14 |
| getContour . . . . .        | 15 |
| getContourRegions . . . . . | 16 |
| getNiche . . . . .          | 17 |
| globalMoran . . . . .       | 18 |
| grid2df . . . . .           | 19 |
| grid2sf . . . . .           | 20 |
| gridDensity . . . . .       | 20 |
| gridSPE . . . . .           | 22 |
| localMoran . . . . .        | 23 |
| mergeROI . . . . .          | 24 |
| normalizeAssay . . . . .    | 25 |
| plotCellCompo . . . . .     | 26 |
| plotContour . . . . .       | 27 |
| plotContourRegion . . . . . | 28 |
| plotCorHeatmap . . . . .    | 29 |
| plotDensCor . . . . .       | 30 |
| plotDensity . . . . .       | 31 |
| plotDR . . . . .            | 32 |

|                                                  |           |
|--------------------------------------------------|-----------|
| plotGrid . . . . .                               | 33        |
| plotImage . . . . .                              | 35        |
| plotLISA . . . . .                               | 35        |
| plotROI . . . . .                                | 36        |
| plotSpatial . . . . .                            | 38        |
| postSelRegion . . . . .                          | 39        |
| realignVisium . . . . .                          | 40        |
| realignVisiumHD . . . . .                        | 40        |
| runPCA . . . . .                                 | 41        |
| runUMAP . . . . .                                | 42        |
| selectRegion . . . . .                           | 43        |
| spe2PB . . . . .                                 | 44        |
| update_bound . . . . .                           | 45        |
| xenium_bc_spe . . . . .                          | 45        |
| [,SpatialExperiment,ANY,ANY,ANY-method . . . . . | 46        |
| <b>Index</b>                                     | <b>47</b> |

---

|                |                                                                    |
|----------------|--------------------------------------------------------------------|
| scider-package | <i>scider: Spatial cell-type inter-correlation by density in R</i> |
|----------------|--------------------------------------------------------------------|

---

## Description

scider is an user-friendly R package providing functions to model the global density of cells in a slide of spatial transcriptomics data. All functions in the package are built based on the SpatialExperiment object, allowing integration into various spatial transcriptomics-related packages from Bioconductor. After modelling density, the package allows for serveral downstream analysis, including colocalization analysis, boundary detection analysis and differential density analysis.

scider implements functions to analyse spatial transcriptomics data with cell type annotations by performing cell type correlation via density estimation and cell type co-localization via real number distance. Functions include density estimation, statistical modelling and visualizations.

## Details

scider uses SpatialExperiment objects as the main infrastructure, which can easily be integrated with a wide variety of Bioconductor packages.

## Author(s)

**Maintainer:** Yunshun Chen <yuchen@wehi.edu.au> ([ORCID](#))

Authors:

- Mengbo Li <li.me@wehi.edu.au> ([ORCID](#))
- Ning Liu <liu.n@wehi.edu.au> ([ORCID](#))
- Quoc Hoang Nguyen <nguyen.q@wehi.edu.au> ([ORCID](#))

Ning Liu <liu.n@wehi.edu.au>, Mengbo Li <li.me@wehi.edu.au>, Yunshun Chen <yuchen@wehi.edu.au>, Quoc Hoang Nguyen <nguyen.q@wehi.edu.au>

**See Also**

Useful links:

- <https://github.com/ChenLaboratory/scider>
- <https://chenlaboratory.github.io/scider/>
- Report bugs at <https://github.com/ChenLaboratory/scider/issues>

---

allocateCells

*Annotate all cells with contour level of cell type-specific density.*

---

**Description**

Annotate all cells with contour level of cell type-specific density.

**Usage**

```
allocateCells(
  spe,
  to.roi = TRUE,
  roi = NULL,
  to.contour = TRUE,
  contour = NULL
)
```

**Arguments**

|            |                                                                                                                                                                                       |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                                                                                                                                           |
| to.roi     | Logical. Whether to allocate cells to ROIs.                                                                                                                                           |
| roi        | Character. The name of the group or cell type on which the roi is computed. If NULL, then the cell allocation will be performed for all detected roi Default to NULL.                 |
| to.contour | Logical. Whether to allocate cells to contour levels.                                                                                                                                 |
| contour    | Character. The name of the group or cell type on which the contour level is computed. If NULL, then the cell allocation will be performed for all detected contours. Default to NULL. |

**Value**

A SpatialExperiment object. An extra column is added to the colData.

**Examples**

```
data("xenium_bc_spe")
spe <- gridDensity(spe)
coi <- "Breast cancer"
spe <- findROI(spe, coi = coi)
spe <- getContour(spe, coi = coi)
spe <- allocateCells(spe, contour = coi)
```

---

|               |                                               |
|---------------|-----------------------------------------------|
| cellsInRegion | <i>Check which cells are in which regions</i> |
|---------------|-----------------------------------------------|

---

**Description**

Check which cells are in which regions

**Usage**

```
cellsInRegion(spe, region, name_to, NA_level = "0", levels = NULL)
```

**Arguments**

|          |                                                                  |
|----------|------------------------------------------------------------------|
| spe      | A SpatialExperiment object.                                      |
| region   | List or an sf object that represents a region or an ROI.         |
| name_to  | Colname in colData(spe) to store the annotation.                 |
| NA_level | Label for cells not falling in any of the regions. Default to 0. |
| levels   | Factor levels.                                                   |

**Value**

A SpatialExperiment object. The region information of each cell is stored in the colData.

---

|                |                                                               |
|----------------|---------------------------------------------------------------|
| computeDensity | <i>Perform kernel density estimation on SpatialExperiment</i> |
|----------------|---------------------------------------------------------------|

---

**Description**

Perform kernel density estimation on SpatialExperiment

**Usage**

```
computeDensity(
  xy,
  kernel = c("gaussian", "epanechnikov", "quartic", "disc"),
  bandwidth = NULL,
  weights = NULL,
  ngrid.x = NULL,
  xlim = NULL,
  ylim = NULL,
  diggle = FALSE,
  gridInfo = FALSE
)
```

**Arguments**

|           |                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| xy        | A numeric matrix of spatial coordinates.                                                                                                        |
| kernel    | The smoothing kernel. Options are gaussian, epanechnikov, quartic or disc.                                                                      |
| bandwidth | The smoothing bandwidth. By default performing automatic bandwidth selection using cross-validation using function spatstat.explore::bw.diggle. |
| weights   | Optional weights to be attached to the points.                                                                                                  |
| ngrid.x   | Number of grids in the x-direction.                                                                                                             |
| xlim      | The range of the x-coordinates of the image.                                                                                                    |
| ylim      | The range of the y-coordinates of the image.                                                                                                    |
| diggle    | Logical. If TRUE, use the Jones-Diggle improved edge correction. See spatstat.explore::density.ppp() for details.                               |
| gridInfo  | Logical. If TRUE, then the grid information is also returned.                                                                                   |

**Value**

Output from spatstat.explore::density.ppp.

---

|                   |                                                               |
|-------------------|---------------------------------------------------------------|
| computeDensityHex | <i>Perform kernel density estimation on SpatialExperiment</i> |
|-------------------|---------------------------------------------------------------|

---

**Description**

Perform kernel density estimation on SpatialExperiment

**Usage**

```
computeDensityHex(
  xy,
  kernel = c("gaussian"),
  bandwidth = NULL,
  weights = NULL,
  ngrid.x = NULL,
  xlim = NULL,
  ylim = NULL,
  diggle = FALSE,
  gridInfo = FALSE
)
```

**Arguments**

|           |                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| xy        | A numeric matrix of spatial coordinates.                                                                                                        |
| kernel    | The smoothing kernel. Options are gaussian, epanechnikov, quartic or disc. ONLY GAUSSIAN IS IMPLEMENTED                                         |
| bandwidth | The smoothing bandwidth. By default performing automatic bandwidth selection using cross-validation using function spatstat.explore::bw.diggle. |
| weights   | Optional weights to be attached to the points.                                                                                                  |
| ngrid.x   | Number of grids in the x-direction.                                                                                                             |
| xlim      | The range of the x-coordinates of the image.                                                                                                    |
| ylim      | The range of the y-coordinates of the image.                                                                                                    |
| diggle    | Logical. If TRUE, use the Jones-Diggle improved edge correction. See spatstat.explore::density.ppp() for details.                               |
| gridInfo  | Logical. If TRUE, then the grid information is also returned.                                                                                   |

**Value**

Output from spatstat.explore::density.ppp.

---

contour2sf

---

*Draw a contour region on some density level*


---

**Description**

Draw a contour region on some density level

**Usage**

```
contour2sf(spe, contour, cutoff)
```

**Arguments**

|         |                                                 |
|---------|-------------------------------------------------|
| spe     | A SpatialExperiment object.                     |
| contour | Name in metadata.                               |
| cutoff  | A numeric scalar specifying the density cutoff. |

**Value**

An sf object of the contour region of the specified level.

---

|            |                                                                     |
|------------|---------------------------------------------------------------------|
| coord_hash | <i>Hash two 15-bytes signed integers into one 32-bytes integer.</i> |
|------------|---------------------------------------------------------------------|

---

**Description**

Hash two 15-bytes signed integers into one 32-bytes integer.

**Usage**

```
coord_hash(a, b)
```

**Arguments**

|      |                                                   |
|------|---------------------------------------------------|
| a, b | Integer vectors of same lengths. Can be negative. |
|------|---------------------------------------------------|

**Details**

Should work for a,b in range  $(-2^{14}, 2^{14}-1)$  which is good enough for our purpose.

Integer in R is 32-bytes but reserve 1 byte for NA

---

|            |                                                             |
|------------|-------------------------------------------------------------|
| corDensity | <i>Test for density correlation between two cell types.</i> |
|------------|-------------------------------------------------------------|

---

**Description**

Test for density correlation between two cell types.

**Usage**

```
corDensity(spe, coi = NULL, roi = NULL, probs = 0.85, trace = FALSE)
```

**Arguments**

|       |                                                                                                                                                                                                                    |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe   | A SpatialExperiment object.                                                                                                                                                                                        |
| coi   | Character vector for cell types of interest for density correlation analysis. Default is NULL, which is to consider all cell types previously calculated in the gridDensity() step.                                |
| roi   | Character. The name of the group or cell type on which the roi is computed. Default is NULL for no subsetting cell types by ROI                                                                                    |
| probs | A numeric scalar. The threshold of proportion that used to filter grids by density when ROIs have not been identified previously. Ignored if 'roi' is present in the 'metadata' component of spe. Default to 0.85. |
| trace | Logical. If TRUE, print the process of testing. Default to FALSE.                                                                                                                                                  |

**Value**

A DataFrame containing the testing results.

**Examples**

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts", "B cells", "T cells")

spe <- gridDensity(spe, coi = coi)

spe <- findROI(spe, coi = coi, method = "walktrap")

result <- corDensity(spe, roi = coi)
```

---

findNbrsGrid

---

Construct a neighbour list from grid coordinates.

---

**Description**

Construct a neighbour list from grid coordinates.

**Usage**

```
findNbrsGrid(
  spe,
  n = 1,
  radius = NULL,
  diagonal = FALSE,
  dist_func = c("idw", "exp", "binary", "none"),
  dist_type = c("euclidean", "manhattan"),
  standardisation = c("row", "none"),
```

```

    scale = 1,
    nbrs_name = NULL,
    cpu_threads = 6
  )

```

### Arguments

|                 |                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe             | A SpatialExperiment object.                                                                                                                           |
| n               | Integer. Search for neighbours within (...). Either the number of neighbors or radius                                                                 |
| radius          | Numeric. Search for neighbours within the radius.                                                                                                     |
| diagonal        | Whether to consider diagonal connection if using square grid                                                                                          |
| dist_func       | Options for distance-based weight. "idw" for inverse distance, "exp" for exponential decay, "binary" for constant weight, and "raw" for raw distance. |
| dist_type       | Options of using euclidean or manhattan for distance calculation                                                                                      |
| standardisation | Options for weight standardisation. "none" for nothing, and "row" for dividing weights by number of neighbours.                                       |
| scale           | Numeric scaler for weight scaling.                                                                                                                    |
| nbrs_name       | Name of the neighbour list to be stored. Default to be "grid".                                                                                        |
| cpu_threads     | Number of cpu threads for parallel computation.                                                                                                       |

### Details

If n is used, distance is scaled to unit distance

### Value

A SpatialExperiment object with neighbour list stored in `spe@metadata$nbrs$grid[[nbrs_name]]`

### Examples

```

data("xenium_bc_spe")
spe <- gridDensity(spe)
spe <- findNbrsGrid(spe,n=3)

```

---

findNbrsSNN

*Construct a SNN neighbour list from assay.*

---

### Description

Construct a SNN neighbour list from assay.

**Usage**

```
findNbrsSNN(
  spe,
  assay = NULL,
  dimred = "PCA",
  n_dimred = 10,
  k = 20,
  BNPARAM = BiocNeighbors::AnnoyParam(),
  type = c("rank", "number", "jaccard"),
  nbrs_name = NULL,
  cpu_threads = 6
)
```

**Arguments**

|             |                                                                                                                                   |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------|
| spe         | A SpatialExperiment object.                                                                                                       |
| assay       | Name of assay for clustering. Incompatible with dimred.                                                                           |
| dimred      | Name of the dimensionality reduction (e.g. PCA) for clustering. Incompatible with assay                                           |
| n_dimred    | Integer scalar or vector specifying the dimensions to use if dimred is specified.                                                 |
| k           | Integer scalar for number of nearest neighbors to find.                                                                           |
| BNPARAM     | <a href="#">BiocNeighborParam</a> object specifying the nearest neighbor algorithm. Default is Annoy.                             |
| type        | Type of weighting scheme for shared neighbors. Options are rank, number, and jaccard. type="rank" is defined in Xu and Su (2015). |
| nbrs_name   | Name of the neighbour list to be stored in spe. Default to be assay/dimred + "_snn".                                              |
| cpu_threads | Number of cpu threads for parallel computation.                                                                                   |

**Details**

Construct a SNN neighbour list using either the spe's assay or reduced dimension and store it in `spe@metadata$nbrs$cell`  
neighbour list contain

**Value**

A spe with the clusters stored in [reducedDims](#).  
A SpatialExperiment object

**Examples**

```
data("xenium_bc_spe")
spe <- runPCA(spe)
spe <- findNbrsSNN(spe, dimred="PCA")
```

---

|                 |                                                                         |
|-----------------|-------------------------------------------------------------------------|
| findNbrsSpatial | <i>Construct a distance-based neighbour list from cell coordinates.</i> |
|-----------------|-------------------------------------------------------------------------|

---

### Description

Construct a distance-based neighbour list from cell coordinates.

### Usage

```
findNbrsSpatial(
  spe,
  k = NULL,
  radius = NULL,
  dist_func = c("idw", "exp", "binary", "none"),
  standardisation = c("none", "row"),
  scale = 1,
  nbrs_name = NULL,
  cpu_threads = 6
)
```

### Arguments

|                 |                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe             | A SpatialExperiment object.                                                                                                                                      |
| k               | Integer scalar for number of nearest neighbours to find. Can be used with radius. See details.                                                                   |
| radius          | Numeric for maximum distance to search for neighbours. Can be with k. See details                                                                                |
| dist_func       | Options for distance-based weight. "idw" for inverse distance, "exp" for exponential decay, "binary" for constant weight, and "none" for raw euclidean distance. |
| standardisation | Options for weight standardisation. "none" for nothing, and "row" for dividing weights by number of neighbours.                                                  |
| scale           | Numeric scaler for weight scaling.                                                                                                                               |
| nbrs_name       | Name of the neighbour list to be stored. Default to be "spatial".                                                                                                |
| cpu_threads     | Number of cpu threads for parallel computation.                                                                                                                  |

### Details

if only k is provided, neighbours are found using [findKNN](#). If only radius is provided, neighbours are found using [findNeighbors](#). If both are provided, then knn is done first then neighbours are filtered to only those within radius.

### Value

A SpatialExperiment object with neighbour list stored in `spe@metadata$nbrs$cell[[nbrs_name]]`

**Examples**

```
data("xenium_bc_spe")
spe <- findNbrsSpatial(spe,k=20,radius=100)
```

---

|         |                                                                                |
|---------|--------------------------------------------------------------------------------|
| findROI | <i>Find ROIs based on cell type-specific densities via graph-based method.</i> |
|---------|--------------------------------------------------------------------------------|

---

**Description**

Find ROIs based on cell type-specific densities via graph-based method.

**Usage**

```
findROI(
  spe,
  coi = NULL,
  probs = 0.85,
  min.density = NULL,
  ngrid.min = 20,
  method = c("greedy", "walktrap", "connected", "hdbscan", "eigen", "dbscan"),
  diag.nodes = FALSE,
  sequential.roi.name = TRUE,
  zoom.in = FALSE,
  zoom.in.size = 500L,
  ...
)
```

**Arguments**

|                     |                                                                                                                                                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe                 | A SpatialExperiment object.                                                                                                                                   |
| coi                 | A character vector of cell types of interest (COIs). Default to all cell types.                                                                               |
| probs               | A numeric scalar. The threshold of proportion that used to filter grids by density. Default to 0.85.                                                          |
| min.density         | A numeric value. The cut-off value used to filter grids by density. Default is NULL and overwrites probs.                                                     |
| ngrid.min           | An integer. The minimum number of grids required for defining a ROI. Default to 20.                                                                           |
| method              | The community detection method to be used, possible options are greedy, walktrap, connected, hdbscan, eigen or dbscan. Default to greedy, can be abbreviated. |
| diag.nodes          | Logical. Set this to TRUE to allow diagonal grid points to be adjacent nodes.                                                                                 |
| sequential.roi.name | Logical. Set this to FALSE if you want the original ROI name before filtering are retained.                                                                   |

|              |                                                                                    |
|--------------|------------------------------------------------------------------------------------|
| zoom.in      | Logical. For very large ROIs, whether to zoom in and try to get more refined ROIs. |
| zoom.in.size | A numeric scaler. Smallest size of an ROI to be able to zoom in. Default is 500L.  |
| ...          | Other parameters that passed to walktrap.community when method = "walktrap".       |

**Value**

A SpatialExperiment object.

**Examples**

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts")

spe <- gridDensity(spe, coi = coi)

spe <- findROI(spe, coi = coi, method = "walktrap")
```

---

|             |                                                  |
|-------------|--------------------------------------------------|
| getClusters | <i>Cluster cells in spe using graph methods.</i> |
|-------------|--------------------------------------------------|

---

**Description**

Cluster cells in spe using graph methods.

**Usage**

```
getClusters(
  spe,
  nbrs_name = NULL,
  method = c("leiden", "louvain"),
  resolution = 1,
  cluster_name = "cluster",
  seed = 1,
  ...
)
```

**Arguments**

|           |                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------|
| spe       | A SpatialExperiment object.                                                                                                              |
| nbrs_name | Name of neighbour list for clustering. If NULL, will use the newest one in spe@metadata\$nbrs\$cell or create one if none are available. |
| method    | Clustering methods. Options are leiden and louvain.                                                                                      |

|              |                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------|
| resolution   | Higher resolution for more clusters and lower for fewer clusters. See <a href="#">cluster_leiden</a> and <a href="#">cluster_louvain</a> |
| cluster_name | Name to store the clusters in spe's <a href="#">colData</a>                                                                              |
| seed         | seed for clustering                                                                                                                      |
| ...          | Other clustering arguments for <a href="#">cluster_leiden</a> or <a href="#">cluster_louvain</a>                                         |

### Details

Cluster cells with igraph using SNN calculated by [findNbrsSNN](#). Any neighbour list in `spe@metadata$nbrs$cell` can also be used

### Value

A spe with the clusters stored in [reducedDims](#).

A SpatialExperiment object

### Examples

```
data("xenium_bc_spe")
spe <- normalizeAssay(spe)
spe <- runPCA(spe)
spe <- findNbrsSNN(spe, dimred="PCA")
spe <- getClusters(spe, resolution=0.5)
```

---

|            |                                 |
|------------|---------------------------------|
| getContour | <i>Get contour from density</i> |
|------------|---------------------------------|

---

### Description

Get contour from density

### Usage

```
getContour(
  spe,
  coi = NULL,
  equal.cell = TRUE,
  bins = NULL,
  binwidth = NULL,
  breaks = NULL,
  id = NULL
)
```

**Arguments**

|            |                                                                                                                                                                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                                                                                                                                                                                                                |
| coi        | A character vector of cell types of interest (COIs). All cell types are chosen if NULL or overall.                                                                                                                                                         |
| equal.cell | Logical. Whether to use produce contour levels so that there are roughly the same number of cells of the COI at each level. Default to TRUE.                                                                                                               |
| bins       | An integer. Number of contour levels.                                                                                                                                                                                                                      |
| binwidth   | A numeric scale of the smoothing bandwidth.                                                                                                                                                                                                                |
| breaks     | A numeric scale referring to the breaks in <code>ggplot2::contour_breaks</code> .                                                                                                                                                                          |
| id         | A character. The name of the column of <code>colData(spe)</code> containing the cell type identifiers. Set to <code>cell_type</code> by default or <code>in_tissue</code> if <code>spe</code> is Visium. Only needed when <code>equal.cell = TRUE</code> . |

**Value**

A SpatialExperiment object. An sf object of the contour region of the specified level is stored in the metadata of the SpatialExperiment object.

**Examples**

```
data("xenium_bc_spe")
spe <- gridDensity(spe)
coi <- "Breast cancer"
spe <- getContour(spe, coi = coi)
```

---

|                   |                                                         |
|-------------------|---------------------------------------------------------|
| getContourRegions | <i>Calculate areas between every two density levels</i> |
|-------------------|---------------------------------------------------------|

---

**Description**

Calculate areas between every two density levels

**Usage**

```
getContourRegions(spe, contour_name)
```

**Arguments**

|              |                                              |
|--------------|----------------------------------------------|
| spe          | A SpatialExperiment object.                  |
| contour_name | Name of contour in <code>spe@metadata</code> |

**Value**

A list of sf objects, each representing the region between two contour density levels.

getNiche

*Build a niche assay based on the profile of neighbouring cells***Description**

Build a niche assay based on the profile of neighbouring cells

**Usage**

```
getNiche(
  spe,
  at = c("cell", "grid"),
  nbrs_name = NULL,
  group.by,
  use_weight = FALSE
)
```

**Arguments**

|            |                                                                                                                                                                           |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object                                                                                                                                                |
| at         | Option of cell or grid neighbourhood                                                                                                                                      |
| nbrs_name  | Name of the neighbour list in spe@metadata\$grid[[at]]                                                                                                                    |
| group.by   | Character vector to group neighbours cell by. Should be in either colData(spe) or spe@metadata\$grid_density, depending on "at". Multiple groups can be used. See details |
| use_weight | Whether to scale each nbr based on its weight                                                                                                                             |

**Details**

For numerical group, result will be sum of nbrs for each cell. For categorical group (factor/string), result will be counts of nbrs belonging in category

**Value**

A matrix where rows are cells/grid points and cols are groups based on group.by

**Examples**

```
data("xenium_bc_spe")

spe <- findNbrsSpatial(spe,k=30)
niche = getNiche(spe,at="cell",group.by="cell_type")
```

---

|             |                                                     |
|-------------|-----------------------------------------------------|
| globalMoran | <i>Calculate global Moran for 1 to 2 variables.</i> |
|-------------|-----------------------------------------------------|

---

### Description

Calculate global Moran for 1 to 2 variables.

### Usage

```
globalMoran(
  spe,
  data1,
  data2 = data1[],
  at = c("grid", "cell"),
  nbrs_name = NULL,
  permutations = 999,
  seed = 123456789,
  cpu_threads = 6
)
```

### Arguments

|              |                                                                                                                     |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| spe          | A SpatialExperiment object.                                                                                         |
| data1        | Numeric vector 1. Must be same length as nrow(spe@metadata\$grid_density) or spatialCoords(spe), depending on 'at'. |
| data2        | Numeric vector 2 for bivariate local Moran. Must be same length as data1.                                           |
| at           | Option of grid or cell for where to look for neighbour list                                                         |
| nbrs_name    | Name of the neighbour list in spe@metadata\$grid[[at]] for Moran's I                                                |
| permutations | Number of permutations for p-value.                                                                                 |
| seed         | Integer. For random permutations.                                                                                   |
| cpu_threads  | (optional) The number of cpu threads used for parallel LISA computation                                             |

### Value

List with global lisa, p.value, and vector of permuted lisa.

### Examples

```
data("xenium_bc_spe")

## At grid.
spe <- gridDensity(spe, coi = "Breast cancer")
dat <- spe@metadata$grid_density$density_breast_cancer
spe <- findNbrsGrid(spe)
res <- globalMoran(spe, data1 = dat, at="grid")
res$lisa
```

```
## At cell.
dat <- as.numeric(spe$cell_type=="Breast cancer")
spe <- findNbrsSpatial(spe,k=10)
res <- globalMoran(spe,data1 = dat,at="cell")
res$lisa
```

---

grid2df

---

*Convert x,y nodes to data.frame of polygons*


---

### Description

Convert x,y nodes to data.frame of polygons

### Usage

```
grid2df(
  spe,
  x = spe@metadata$grid_density$node_x,
  y = spe@metadata$grid_density$node_y,
  reverseY = FALSE,
  ...
)
```

### Arguments

|          |                                                                                                                  |
|----------|------------------------------------------------------------------------------------------------------------------|
| spe      | A SpatialExperiment object with grid density calculated                                                          |
| x        | vector of x nodes of the polygons                                                                                |
| y        | vector of y nodes of the polygons                                                                                |
| reverseY | Reverse y coordinates. Can be numeric to specify the value to subtract y coordinates from (reverseY - y coords). |
| ...      | other elements to be stored as columns of the data.frame. Each one must be a vector same length as x.            |

### Details

Basically grid2sf() but returns a data.frame for plotting with geom\_polygon(), which allows for scale\_\*\_transform(), unlike geom\_sf().

Column names are kept similar to sf::st\_coordinates

### Value

data.frame with X, Y, and L2. Points with the same L2 belong to the same polygons

---

|         |                                         |
|---------|-----------------------------------------|
| grid2sf | <i>Convert x,y nodes to sf polygons</i> |
|---------|-----------------------------------------|

---

### Description

Convert x,y nodes to sf polygons

### Usage

```
grid2sf(
  spe,
  x = spe@metadata$grid_density$node_x,
  y = spe@metadata$grid_density$node_y,
  reverseY = FALSE
)
```

### Arguments

|          |                                                                                                                  |
|----------|------------------------------------------------------------------------------------------------------------------|
| spe      | A SpatialExperiment object with grid density calculated                                                          |
| x        | vector of x nodes of the polygons                                                                                |
| y        | vector of y nodes of the polygons                                                                                |
| reverseY | Reverse y coordinates. Can be numeric to specify the value to subtract y coordinates from (reverseY - y coords). |

### Details

Default is to generate sf polygons for all grid. For plotting with geom\_sf, use sf::st\_as\_sf(grid2sf2(spe)) to convert list into Geometry Set.

### Value

List of sf polygons

---

|             |                                                                                          |
|-------------|------------------------------------------------------------------------------------------|
| gridDensity | <i>Perform kernel density estimation on SpatialExperiment for cell types of interest</i> |
|-------------|------------------------------------------------------------------------------------------|

---

### Description

Perform kernel density estimation on SpatialExperiment for cell types of interest

**Usage**

```
gridDensity(
  spe,
  id = if (isVisium) NULL else "cell_type",
  coi = NULL,
  feature = NULL,
  assay = "counts",
  kernel = "gaussian",
  bandwidth = NULL,
  ngrid.x = NULL,
  grid.length.x = NULL,
  diggle = FALSE,
  grid.type = c("hex", "square"),
  isVisium = FALSE,
  filterToVisiumSpot = isVisium
)
```

**Arguments**

|                                 |                                                                                                                                                                                                |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>spe</code>                | A SpatialExperiment object.                                                                                                                                                                    |
| <code>id</code>                 | A character. The name of the column of <code>colData(spe)</code> containing the cell type identifiers. Set to <code>cell_type</code> by default. Set to <code>NULL</code> for overall density. |
| <code>coi</code>                | A character vector of cell types of interest (COIs). Default to all cell types.                                                                                                                |
| <code>feature</code>            | Feature(s) to calculate density with. Must be in <code>rownames(spe)</code> .                                                                                                                  |
| <code>assay</code>              | Name of assay to use for finding feature(s).                                                                                                                                                   |
| <code>kernel</code>             | The smoothing kernel. Options are "gaussian", "epanechnikov", "quartic" or "disc". For hexagonal grid, only Gaussian is implemented                                                            |
| <code>bandwidth</code>          | The smoothing bandwidth. By default performing automatic bandwidth selection using cross-validation using function <code>spatstat.explore::bw.diggle</code> .                                  |
| <code>ngrid.x</code>            | Number of grids in the x-direction. Ignored when 'grid.length.x' is specified. Default to <code>NULL</code> .                                                                                  |
| <code>grid.length.x</code>      | Grid length in the x-direction. If both 'ngrid.x' and 'grid.length.x' are <code>NULL</code> , then 'grid.length.x' is set to 100 (micron) by default.                                          |
| <code>diggle</code>             | Logical. If <code>TRUE</code> , use the Jones-Diggle improved edge correction. See <code>spatstat.explore::density.ppp()</code> for details.                                                   |
| <code>grid.type</code>          | Type of grid can be either hexagon or square.                                                                                                                                                  |
| <code>isVisium</code>           | Logical. If <code>TRUE</code> , fit hexagonal grids to Visium spots by replacing spatial coords with array rows & array cols.                                                                  |
| <code>filterToVisiumSpot</code> | Logical. If <code>TRUE</code> , filter grid polygons to only those with a Visium spot underneath.                                                                                              |

**Value**

A SpatialExperiment object. Grid density estimates for all cell type of interest are stored in `spe@metadata$grid_density`. Grid information is stored in `spe@metadata$grid_info`

**Examples**

```
data("xenium_bc_spe")

spe <- gridDensity(spe)
```

---

gridSPE

*Summarize a SpatialExperiment object at grid-level*


---

**Description**

Summarize a SpatialExperiment object at grid-level

**Usage**

```
gridSPE(spe, cell.count = FALSE, id = "cell_type", split.count.by = id)
```

**Arguments**

|                |                                                                                                                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe            | A SpatialExperiment object.                                                                                                                                                                                                              |
| cell.count     | Logical. Whether to obtain the number of cells within each group identified by the 'id' column in colData(spe). Default to FALSE.                                                                                                        |
| id             | A character. The name of the column of colData(spe) containing the cell type identifiers. Set to 'cell_type' by default.                                                                                                                 |
| split.count.by | A character. The name of the column of colData(spe). When it is not NULL, a grid-level count matrix is calculated for each member specified in that column of colData(spe) and stored in the assays(spe). Set to 'cell_type' by default. |

**Value**

A SpatialExperiment object.

**Examples**

```
data("xenium_bc_spe")

spe <- gridDensity(spe)

spe_grid <- gridSPE(spe)
```

---

|            |                                                    |
|------------|----------------------------------------------------|
| localMoran | <i>Calculate local Moran for 1 to 2 variables.</i> |
|------------|----------------------------------------------------|

---

## Description

Calculate local Moran for 1 to 2 variables.

## Usage

```
localMoran(
  spe,
  data1,
  data2 = data1[],
  at = c("grid", "cell"),
  nbrs_name = NULL,
  hhonly = FALSE,
  significance_cutoff = 0.05,
  permutations = 999,
  seed = 123456789,
  cpu_threads = 6
)
```

## Arguments

|                     |                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| spe                 | A SpatialExperiment object.                                                                                                           |
| data1               | Numeric vector 1. Must be same length as number of grid points or cells, depending on 'at'.                                           |
| data2               | Numeric vector 2 for bivariate local Moran. Must be same length as data1.                                                             |
| at                  | Option of grid or cell for where to look for neighbour list                                                                           |
| nbrs_name           | Name of the neighbour list in <code>spe@metadata\$grid[[at]]</code> for Moran's I. If NULL, will use the newest neighbour list.       |
| hhonly              | Only high-high clusters, which is more interpretable. Other clusters (e.g. "Low-Low", "High-Low", ...) will be assigned as undefined. |
| significance_cutoff | Cutoff for p-value to filter non-significant clusters                                                                                 |
| permutations        | Number of permutations for p-value.                                                                                                   |
| seed                | Integer. For random permutations.                                                                                                     |
| cpu_threads         | The number of cpu threads used for parallel computation.                                                                              |

## Value

List of `lisa_value`, `clusters`, and pseudo p-value.

## Examples

```
data("xenium_bc_spe")

## At grid.
spe <- gridDensity(spe, coi = "Breast cancer")
dat <- spe@metadata$grid_density$density_breast_cancer
spe <- findNbrsGrid(spe)
res <- localMoran(spe, data1=dat, at = "grid")

## At cell.
dat <- as.numeric(spe$cell_type=="Breast cancer")
spe <- findNbrsSpatial(spe, k=20)
res <- localMoran(spe, data1=dat, at="cell")
```

---

mergeROI

*Manually merge ROIs*


---

## Description

Manually merge ROIs

## Usage

```
mergeROI(
  spe,
  roi = NULL,
  merge.list = NULL,
  remove.ids = NULL,
  id = "component",
  rename = FALSE
)
```

## Arguments

|            |                                                                                                                                                                                                                      |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                                                                                                                                                                          |
| roi        | Character. The name of the group or cell type on which the roi is computed. All cell types are chosen if NULL or 'overall'.                                                                                          |
| merge.list | A (named) list of vectors of ROI ids to be merged. Each vector in the list should be of length greater than or equal to 2. If no name is specified, the merged ROI will be named by concatenating ROIs being merged. |
| remove.ids | Optional. A vector of ROI ids to be removed.                                                                                                                                                                         |
| id         | Character. The name of the column in spe@metadata\$roi that stores the ROIs to be merged. Default is "component".                                                                                                    |
| rename     | Logical. If TRUE, names of merge.list are ignored. ROIs will be given a new name. For the unmerged ROIs, their new names are not necessarily the same as those before merging.                                       |

**Value**

A SpatialExperiment object.

**Examples**

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts")

spe <- gridDensity(spe, coi = coi)

spe <- findROI(spe, coi = coi, method = "walktrap")

spe <- mergeROI(spe, roi = coi, list("1-2" = 1:2))
```

---

|                |                                             |
|----------------|---------------------------------------------|
| normalizeAssay | <i>Perform log normalization for counts</i> |
|----------------|---------------------------------------------|

---

**Description**

Perform log normalization for counts

**Usage**

```
normalizeAssay(
  spe,
  transformation = c("log"),
  scale.factor = 1e+06,
  assay = "counts",
  name = "logcounts"
)
```

**Arguments**

|                |                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------|
| spe            | A SpatialExperiment object.                                                                               |
| transformation | Choice of transformation. "Log" for log1p                                                                 |
| scale.factor   | Factor to multiply the count of each cell by. A single value or a numeric vector equal to number of cells |
| assay          | Name of assay in spe to perform the transformation on                                                     |
| name           | Name of the transformed assay                                                                             |

**Value**

A SpatialExperiment object

**Examples**

```
data("xenium_bc_spe")
spe <- normalizeAssay(spe)
```

---

plotCellCompo

---

*Plot cell type composition in each density level of cell of interest.*


---

**Description**

Plot cell type composition in each density level of cell of interest.

**Usage**

```
plotCellCompo(
  spe,
  contour = NULL,
  id = "cell_type",
  roi = NULL,
  self.included = TRUE
)
```

**Arguments**

|               |                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe           | A SpatialExperiment object.                                                                                                                                    |
| contour       | A character vector of cell type(s) on which the contour density level is calculated. If NULL, it looks for 'overall_contour' in colData(spe). Default to NULL. |
| id            | A character. The name of the column of colData(spe) containing the cell type identifiers. Set to 'cell_type' by default.                                       |
| roi           | Character. The name of the group or cell type on which the roi is computed. Default is NULL for no plotting by ROI                                             |
| self.included | Logical. Whether to include all the cell types in the plot. Default to TRUE. If FALSE, the cell types specified in 'contour' will not be included in the plot. |

**Value**

A ggplot object.

**Examples**

```
data("xenium_bc_spe")
coi <- "Breast cancer"
spe <- gridDensity(spe, coi = coi)
spe <- findROI(spe, coi = coi)
spe <- getContour(spe, coi = coi)
spe <- allocateCells(spe, contour = coi)
plotCellCompo(spe, contour = "Breast cancer")
```

```
plotCellCompo(spe, contour = "Breast cancer", roi = coi)
```

---

plotContour

*Plot contour lines.*


---

## Description

Plot contour lines.

## Usage

```
plotContour(
  spe,
  coi = NULL,
  overlay = c("cell", "density", "none"),
  id = "cell_type",
  sub.level = NULL,
  line.type = 1,
  line.width = 0.5,
  line.alpha = 1,
  reverseY = NULL,
  ...
)
```

## Arguments

|            |                                                                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                                                                                                        |
| coi        | A character vector of cell types of interest (COIs). All cell types are chosen if NULL or 'overall'.                                               |
| overlay    | Character vector. Options are 'cell' (plot overlay on cells), 'density' (overlay on density), or 'none'. Default to 'cell'.                        |
| id         | A character. The name of the column of colData(spe) containing the cell type identifiers. Set to 'cell_type' by default.                           |
| sub.level  | Character vector. Subset on specific level.                                                                                                        |
| line.type  | shape of contour. See 'ggplot2::geom_path()'.                                                                                                      |
| line.width | size of contour.                                                                                                                                   |
| line.alpha | alpha of contour between 0 and 1.                                                                                                                  |
| reverseY   | Logical. Whether to reverse Y coordinates. Default is TRUE if the spe contains an image (even if not plotted) and FALSE if otherwise.              |
| ...        | Aesthetic mappings to pass to <a href="#">plotSpatial</a> , <a href="#">plotDensity</a> , or <a href="#">plotImage</a> , depending on the overlay. |

## Value

A ggplot object.

**Examples**

```
data("xenium_bc_spe")

spe <- gridDensity(spe)

coi <- "Breast cancer"

spe <- getContour(spe, coi = coi)

plotContour(spe, coi = coi, line.width = 0.3, pt.alpha = 0.2)
```

---

|                   |                                                                       |
|-------------------|-----------------------------------------------------------------------|
| plotContourRegion | <i>Visualising an sf object (for internal use only at the moment)</i> |
|-------------------|-----------------------------------------------------------------------|

---

**Description**

Visualising an sf object (for internal use only at the moment)

**Usage**

```
plotContourRegion(
  spe,
  coi,
  id = "cell_type",
  overlay = c("density", "cell"),
  sub.level
)
```

**Arguments**

|           |                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe       | A SpatialExperiment object.                                                                                                                                                                            |
| coi       | A character vector of length 1 of the cell type of interest.                                                                                                                                           |
| id        | A character. The name of the column of colData(spe) containing the cell type identifiers. Set to cell_type by default.                                                                                 |
| overlay   | Character vector. Either plot overlay on density or cells.                                                                                                                                             |
| sub.level | Numeric vector of length 1 or 2, identifies which density level to plot. When length is 1, plot the density region above this level. When length is 2, plot the density region between the two levels. |

**Value**

A ggplot object.

---

|                |                                             |
|----------------|---------------------------------------------|
| plotCorHeatmap | <i>Plot model statistics using heatmap.</i> |
|----------------|---------------------------------------------|

---

## Description

Plot model statistics using heatmap.

## Usage

```
plotCorHeatmap(
  model.result,
  stats = c("cor.coef", "t", "p.Pos", "p.Neg"),
  roi = "all",
  cell.type = "all",
  silent = FALSE
)
```

## Arguments

|              |                                                                          |
|--------------|--------------------------------------------------------------------------|
| model.result | A dataFrame object.                                                      |
| stats        | Character value. Choose either coefficient or t. Coefficient by default. |
| roi          | Character value. By default is all. The specific ROIs to be plotted.     |
| cell.type    | Character value. By default is all. The cell types to be plotted.        |
| silent       | Do not draw the plot (useful when using the gtable output).              |

## Value

A heatmap object.

## Examples

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts", "B cells", "T cells")

spe <- gridDensity(spe, coi = coi)

spe <- findROI(spe, coi = coi)

model_result <- corDensity(spe, roi = coi)

plotCorHeatmap(model_result$ROI)
```

---

|             |                                                        |
|-------------|--------------------------------------------------------|
| plotDensCor | <i>Plot density correlation between two cell types</i> |
|-------------|--------------------------------------------------------|

---

### Description

Plot density correlation between two cell types

### Usage

```
plotDensCor(
  spe,
  celltype1 = NULL,
  celltype2 = NULL,
  roi = NULL,
  probs = 0.85,
  fit = c("spline", "linear"),
  df = 3,
  pt.shape = 21,
  pt.size = 1.5,
  pt.alpha = 1,
  line.type = 1,
  line.width = 1,
  line.alpha = 1
)
```

### Arguments

|            |                                                                                                                                                                                                                    |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                                                                                                                                                                        |
| celltype1  | Cell type 1 to compare.                                                                                                                                                                                            |
| celltype2  | Cell type 2 to compare.                                                                                                                                                                                            |
| roi        | Character. The name of the group or cell type on which the roi is computed. Default is NULL for no facetting by ROI                                                                                                |
| probs      | A numeric scalar. The threshold of proportion that used to filter grids by density when ROIs have not been identified previously. Ignored if 'roi' is present in the 'metadata' component of spe. Default to 0.85. |
| fit        | Character. Options are "spline" and "linear".                                                                                                                                                                      |
| df         | Integer. Degrees of freedom of the spline fit. Default to 3 (i.e., a cubic spline fit).                                                                                                                            |
| pt.shape   | shape of points.                                                                                                                                                                                                   |
| pt.size    | size of points.                                                                                                                                                                                                    |
| pt.alpha   | alpha of points between 0 and 1.                                                                                                                                                                                   |
| line.type  | shape of line.                                                                                                                                                                                                     |
| line.width | size of line.                                                                                                                                                                                                      |
| line.alpha | alpha of line between 0 and 1.                                                                                                                                                                                     |

**Value**

A ggplot object.

**Examples**

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts")

spe <- gridDensity(spe, coi = coi)

spe <- findROI(spe, coi = coi, method = "walktrap")

plotDensCor(spe, celltype1 = "Breast cancer", celltype2 = "Fibroblasts", roi = coi)
```

---

|             |                                 |
|-------------|---------------------------------|
| plotDensity | <i>Plot grid-based density.</i> |
|-------------|---------------------------------|

---

**Description**

Plot grid-based density.

**Usage**

```
plotDensity(spe, coi = NULL, probs = 0.5, reverseY = NULL, ...)
```

**Arguments**

|          |                                                                                                                                       |
|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| spe      | A SpatialExperiment object.                                                                                                           |
| coi      | A character vector of cell types of interest (COIs) to be plotted. Default to all cell types.                                         |
| probs    | Numeric value between 0 and 1, used for filtering uninformative grid, default is 0.5.                                                 |
| reverseY | Logical. Whether to reverse Y coordinates. Default is TRUE if the spe contains an image (even if not plotted) and FALSE if otherwise. |
| ...      | Parameters pass to <a href="#">plotGrid</a>                                                                                           |

**Value**

A ggplot object.

**Examples**

```
data("xenium_bc_spe")

spe <- gridDensity(spe)

plotDensity(spe, coi = "Breast cancer")

plotDensity(spe, coi = "Fibroblasts")
```

---

|        |                                 |
|--------|---------------------------------|
| plotDR | <i>Plot reduced dimensions.</i> |
|--------|---------------------------------|

---

**Description**

plotDR is the main function for plotting reduced dimension. Others are wrapper functions for convenience.

**Usage**

```
plotDR(
  spe,
  dimred = NULL,
  dims = c(1, 2),
  group.by = NULL,
  feature = NULL,
  assay = "counts",
  cols = NULL,
  pt.shape = 16,
  pt.size = 1,
  pt.alpha = 0.6,
  label = NULL,
  label.x = NULL,
  label.y = NULL,
  cols.scale = NULL
)

plotUMAP(spe, dimred = "UMAP", ...)

plotPCA(spe, dimred = "PCA", ...)
```

**Arguments**

|        |                                                                                           |
|--------|-------------------------------------------------------------------------------------------|
| spe    | A SpatialExperiment object.                                                               |
| dimred | Name of the reduced dimension in <a href="#">reducedDims</a>                              |
| dims   | Numeric vector length 2 for the dimensions to be plotted. Default to first two dimensions |

|            |                                                                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| group.by   | values to group points by. Must be in colData of spe. If NULL, will try with 'cols' if available.                                                          |
| feature    | Feature to group polygons by. Must be in rownames(spe).                                                                                                    |
| assay      | Name of assay to use for plotting feature.                                                                                                                 |
| cols       | Colour palette. Can be a vector of colours or a function that accepts an integer n and return n colours.                                                   |
| pt.shape   | shape of points.                                                                                                                                           |
| pt.size    | size of points.                                                                                                                                            |
| pt.alpha   | alpha of points between 0 and 1.                                                                                                                           |
| label      | label for the legend                                                                                                                                       |
| label.x    | label for the x-axis                                                                                                                                       |
| label.y    | label for the y-axis                                                                                                                                       |
| cols.scale | vector of position for color if colors should not be evenly positioned. See <a href="#">scale_color_gradientn</a> . Only applicable for continuous values. |
| ...        | Additional arguments pass to plotDR                                                                                                                        |

**Value**

A ggplot object.

**Examples**

```
data("xenium_bc_spe")
spe = runUMAP(spe)
plotDR(spe, group.by = "cell_type")
```

---

|          |                                 |
|----------|---------------------------------|
| plotGrid | <i>Plot grid from metadata.</i> |
|----------|---------------------------------|

---

**Description**

Plot grid from metadata.

**Usage**

```
plotGrid(
  spe,
  group.by = NULL,
  feature = NULL,
  assay = "counts",
  type = c("raw", "log", "cpm", "logcpm"),
  cols = NULL,
  pol.border = FALSE,
```

```

    pol.alpha = 1,
    probs = 0,
    cutoff = NULL,
    label = NULL,
    cols.scale = NULL,
    reverseY = NULL,
    ...
  )

```

## Arguments

|                         |                                                                                                                                                                                                                  |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>spe</code>        | A SpatialExperiment object.                                                                                                                                                                                      |
| <code>group.by</code>   | values to group polygons by. Must be in <code>spe@metadata\$grid_density</code> , or <code>colData(spe)</code> if <code>gridLevelAnalysis</code> is TRUE. If NULL, will try with <code>cols</code> if available. |
| <code>feature</code>    | Feature to group polygons by. Must be in <code>rownames(spe)</code> .                                                                                                                                            |
| <code>assay</code>      | Name of assay to use for plotting feature.                                                                                                                                                                       |
| <code>type</code>       | Transformation to apply for the group/feature. Options are "raw", "log", "cpm", "logcpm", or a function that accepts and returns a vector of the same length.                                                    |
| <code>cols</code>       | Colour palette. Can be a vector of colours or a function that accepts an integer <code>n</code> and return <code>n</code> colours.                                                                               |
| <code>pol.border</code> | Boolean. Whether to draw border for each polygon.                                                                                                                                                                |
| <code>pol.alpha</code>  | alpha of points between 0 and 1.                                                                                                                                                                                 |
| <code>probs</code>      | Numeric value between 0 and 1, used for filtering uninformative grid. Only applicable for continuous values.                                                                                                     |
| <code>cutoff</code>     | Numeric. Either a vector of length 2 for the lower & upper bounds of data to be included, or length 1 for the lower bound. Override <code>probs</code> if specified. Only applicable for continuous values.      |
| <code>label</code>      | label for the legend                                                                                                                                                                                             |
| <code>cols.scale</code> | vector of position for color if colors should not be evenly positioned. See <a href="#">scale_fill_gradientn</a> . Only applicable for continuous values.                                                        |
| <code>reverseY</code>   | Logical. Whether to reverse Y coordinates. Default is TRUE if the <code>spe</code> contains an image (even if not plotted) and FALSE if otherwise.                                                               |
| <code>...</code>        | Parameters pass to <a href="#">plotImage</a>                                                                                                                                                                     |

## Value

A ggplot object.

## Examples

```

data("xenium_bc_spe")

spe <- gridDensity(spe)

plotGrid(spe, group.by = "density_overall")

```

---

|           |                                     |
|-----------|-------------------------------------|
| plotImage | <i>Plot background image of spe</i> |
|-----------|-------------------------------------|

---

**Description**

Plot background image of spe

**Usage**

```
plotImage(
  spe,
  image = TRUE,
  image_id = NULL,
  sample_id = NULL,
  reverseY = NULL,
  crop = TRUE,
  image.alpha = 1
)
```

**Arguments**

|                     |                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| spe                 | A SpatialExperiment object.                                                                                                           |
| image               | Logical. Whether to plot image (if present).                                                                                          |
| image_id, sample_id | sample and image identifiers for scaling factor. See <a href="#">scaleFactors</a> for default behavior.                               |
| reverseY            | Logical. Whether to reverse Y coordinates. Default is TRUE if the spe contains an image (even if not plotted) and FALSE if otherwise. |
| crop                | Whether to crop the plot to the spots.                                                                                                |
| image.alpha         | alpha of points between 0 and 1.                                                                                                      |

**Value**

a ggplot object if there is a valid image, else NULL.

---

|          |                                   |
|----------|-----------------------------------|
| plotLISA | <i>Plotting LISA (e.g. moran)</i> |
|----------|-----------------------------------|

---

**Description**

Plotting LISA (e.g. moran)

**Usage**

```
plotLISA(
  spe,
  lisa,
  overlay = c("grid", "point"),
  type = c("cluster", "logpvalue"),
  reverseY = NULL,
  ...
)
```

**Arguments**

|          |                                                                                                                                       |
|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| spe      | A SpatialExperiment object.                                                                                                           |
| lisa     | Output from localMoran                                                                                                                |
| overlay  | Option of grid or point. Depend on whether <a href="#">localMoran</a> is calculated at grid or point.                                 |
| type     | Option of cluster or logpvalue for plotting lisa's cluster or p-value, respectively.                                                  |
| reverseY | Logical. Whether to reverse Y coordinates. Default is TRUE if the spe contains an image (even if not plotted) and FALSE if otherwise. |
| ...      | Parameters pass to <a href="#">plotGrid</a> or <a href="#">plotSpatial</a> , depending on overlay.                                    |

**Value**

a ggplot object

**Examples**

```
data("xenium_bc_spe")
spe <- gridDensity(spe, coi = "Breast cancer")
dat <- spe@metadata$grid_density$density_breast_cancer
spe <- findNbrsGrid(spe)
res <- localMoran(spe, data1=dat, at = "grid")
plotLISA(spe, lisa = res)
```

---

plotROI

*Plot ROIs on spatial.*


---

**Description**

Plot ROIs on spatial.

**Usage**

```
plotROI(
  spe,
  roi = NULL,
  id = "cell_type",
  label = TRUE,
  show.legend = FALSE,
  reverseY = NULL,
  ...
)
```

**Arguments**

|                          |                                                                                                                                                                              |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>spe</code>         | A <code>SpatialExperiment</code> object.                                                                                                                                     |
| <code>roi</code>         | Character. The name of the group or cell type on which the roi is computed. All cell types are chosen if <code>NULL</code> or <code>'overall'</code> .                       |
| <code>id</code>          | Character. The name of the column of <code>colData(spe)</code> containing the cell type identifiers. Set to <code>cell_type</code> by default.                               |
| <code>label</code>       | Logical. Show ROI label or not.                                                                                                                                              |
| <code>show.legend</code> | Logical. Show legend or not.                                                                                                                                                 |
| <code>reverseY</code>    | Logical. Whether to reverse Y coordinates. Default is <code>TRUE</code> if the <code>spe</code> contains an image (even if not plotted) and <code>FALSE</code> if otherwise. |
| <code>...</code>         | Parameters pass to <a href="#">plotSpatial</a>                                                                                                                               |

**Value**

A `ggplot` object.

**Examples**

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts")

spe <- gridDensity(spe, coi = coi)

spe <- findROI(spe, coi = coi, method = "walktrap", steps = 5)

plotROI(spe, roi = coi, pt.size = 0.3, pt.alpha = 0.2)
```

---

|             |                                                 |
|-------------|-------------------------------------------------|
| plotSpatial | <i>Plot cells based on spatial coordinates.</i> |
|-------------|-------------------------------------------------|

---

## Description

Plot cells based on spatial coordinates.

## Usage

```
plotSpatial(
  spe,
  group.by = NULL,
  feature = NULL,
  assay = "counts",
  type = c("raw", "log", "cpm", "logcpm"),
  cols = NULL,
  pt.shape = 16,
  pt.size = 0.3,
  pt.alpha = 0.5,
  label = NULL,
  cols.scale = NULL,
  reverseY = NULL,
  ...
)
```

## Arguments

|            |                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                                                                                                                   |
| group.by   | values to group points by. Must be in colData of spe. If NULL, will try with 'cols' if available.                                                             |
| feature    | Feature to group polygons by. Must be in rownames(spe).                                                                                                       |
| assay      | Name of assay to use for plotting feature.                                                                                                                    |
| type       | Transformation to apply for the group/feature. Options are "raw", "log", "cpm", "logcpm", or a function that accepts and returns a vector of the same length. |
| cols       | Colour palette. Can be a vector of colours or a function that accepts an integer n and return n colours.                                                      |
| pt.shape   | shape of points.                                                                                                                                              |
| pt.size    | size of points.                                                                                                                                               |
| pt.alpha   | alpha of points between 0 and 1.                                                                                                                              |
| label      | label for the legend                                                                                                                                          |
| cols.scale | vector of position for color if colors should not be evenly positioned. See <a href="#">scale_color_gradientn</a> . Only applicable for continuous values.    |
| reverseY   | Logical. Whether to reverse Y coordinates. Default is TRUE if the spe contains an image (even if not plotted) and FALSE if otherwise.                         |
| ...        | Parameters pass to plotImage                                                                                                                                  |

**Value**

A ggplot object.

**Examples**

```
data("xenium_bc_spe")

plotSpatial(spe, group.by = "cell_type", pt.size = 0.5, pt.alpha = 0.6)
```

---

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| postSelRegion | <i>Merge sel_region from the selectRegion function to SpatialExperiment.</i> |
|---------------|------------------------------------------------------------------------------|

---

**Description**

Merge sel\_region from the selectRegion function to SpatialExperiment.

**Usage**

```
postSelRegion(spe, sel_region)
```

**Arguments**

|            |                                                                  |
|------------|------------------------------------------------------------------|
| spe        | A SpatialExperiment object.                                      |
| sel_region | A dataframe object. Can be generated from function selectRegion. |

**Value**

A SpatialExperiment object.

**Examples**

```
data("xenium_bc_spe")

coi <- c("Breast cancer", "Fibroblasts", "B cells", "T cells")

spe <- gridDensity(spe, coi = coi)

sel_region <- data.frame(
  "node" = seq(10),
  "node_x" = seq(10),
  "node_y" = seq(10)
)

spe1 <- postSelRegion(spe, sel_region)
```

---

|               |                                                    |
|---------------|----------------------------------------------------|
| realignVisium | <i>Scale and straighten out Visium coordinates</i> |
|---------------|----------------------------------------------------|

---

**Description**

Scale and straighten out Visium coordinates

**Usage**

```
realignVisium(spe, distPoint = 100)
```

**Arguments**

|           |                                           |
|-----------|-------------------------------------------|
| spe       | A SpatialExperiment object.               |
| distPoint | Numeric. Desired point to point distance. |

**Details**

This function rescale the distance between points to 100um (or other value) to match the real distance. In addition, Visium spots have a slight tilt to them which this function will also fix

---

|                 |                                                      |
|-----------------|------------------------------------------------------|
| realignVisiumHD | <i>Scale and straighten out VisiumHD coordinates</i> |
|-----------------|------------------------------------------------------|

---

**Description**

Scale and straighten out VisiumHD coordinates

**Usage**

```
realignVisiumHD(spe, distPoint = NULL)
```

**Arguments**

|           |                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------|
| spe       | A SpatialExperiment object.                                                                                 |
| distPoint | Numeric. Desired point to point distance. If NULL, will try to determine the bin level of spe and use that. |

**Details**

This function rescale the distance between points to 8um (or other value) to match the real distance. In addition, Visium spots have a slight tilt to them which this function will also fix

---

|        |                              |
|--------|------------------------------|
| runPCA | <i>Fast PCA using irlba.</i> |
|--------|------------------------------|

---

## Description

Fast PCA using irlba.

## Usage

```
runPCA(  
  spe,  
  n_pcs = 50,  
  assay = "logcounts",  
  centre = TRUE,  
  scale = TRUE,  
  name = "PCA",  
  ...  
)
```

## Arguments

|        |                                                                |
|--------|----------------------------------------------------------------|
| spe    | A SpatialExperiment object.                                    |
| n_pcs  | Number of principal components to calculate                    |
| assay  | Name of assay used for PCA. See details for defaults.          |
| centre | Logical. Whether to centre the assay before PCA.               |
| scale  | Logical. Whether to scale the variance to 1 before PCA.        |
| name   | Name to store the PCA in the spe's <a href="#">reducedDims</a> |
| ...    | Other parameters to be passed to <a href="#">irlba</a> .       |

## Details

By default, runPCA uses logcounts assay (from [normalizeAssay](#)). If that's unavailable, it falls back to counts assay

## Value

A SpatialExperiment with the PCA stored in [reducedDims](#).

## Examples

```
data("xenium_bc_spe")  
  
spe <- runPCA(spe)
```

runUMAP

*UMAP using uwot. Parameters are set to be similar to Seurat's***Description**

UMAP using uwot. Parameters are set to be similar to Seurat's

**Usage**

```
runUMAP(
  spe,
  n_neighbors = 30,
  n_components = 2,
  metric = "cosine",
  min_dist = 0.3,
  assay = NULL,
  dimred = "PCA",
  n_dimred = NULL,
  name = "UMAP",
  ...
)
```

**Arguments**

|                                             |                                                                                   |
|---------------------------------------------|-----------------------------------------------------------------------------------|
| spe                                         | A SpatialExperiment object.                                                       |
| n_neighbors, n_components, metric, min_dist | See <a href="#">umap</a>                                                          |
| assay                                       | Name of assay for UMAP. Incompatible with dimred.                                 |
| dimred                                      | Name of the dimensionality reduction (e.g. PCA) for UMAP. Incompatible with assay |
| n_dimred                                    | Integer scalar or vector specifying the dimensions to use if dimred is specified. |
| name                                        | Name to store the UMAP in the spe's <a href="#">reducedDims</a> .                 |
| ...                                         | Other parameters to be passed to <a href="#">umap</a> .                           |

**Details**

By default, runUMAP uses PCA (from [runPCA](#)). If that's unavailable, it falls back to logcounts, then counts assay.

**Value**

A SpatialExperiment with the UMAP stored in [reducedDims](#).

**Examples**

```
data("xenium_bc_spe")
spe <- runPCA(spe)
spe <- runUMAP(spe,dimred="PCA",n_dimred=10)
```

---

|              |                                            |
|--------------|--------------------------------------------|
| selectRegion | <i>Select region of interest from plot</i> |
|--------------|--------------------------------------------|

---

**Description**

Select region of interest from plot

**Usage**

```
selectRegion(data, x.col = "x", y.col = "y")
```

**Arguments**

|       |                                   |
|-------|-----------------------------------|
| data  | A data.frame object.              |
| x.col | Column name of the x coordinates. |
| y.col | Column name of the y coordinates. |

**Value**

A data.frame object in the global environment.

**Examples**

```
data("xenium_bc_spe")

spe_b <- spe[, SummarizedExperiment::colData(spe)$cell_type == "B cells"]

dat <- as.data.frame(SpatialExperiment::spatialCoords(spe_b))

# selectRegion(dat, x.col = "x_centroid", y.col = "y_centroid")
```

spe2PB

*Given a 'SpatialExperiment' data object, create pseudo-bulk samples using the colData information and return a DGEList object*

## Description

Given a 'SpatialExperiment' data object, create pseudo-bulk samples using the colData information and return a DGEList object

## Usage

```
spe2PB(
  spe,
  by.group = TRUE,
  group.id = "cell_type",
  keep.groups = NULL,
  roi = NULL,
  roi.only = TRUE,
  contour = NULL
)
```

## Arguments

|             |                                                                                                                                                                                  |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| spe         | A SpatialExperiment object.                                                                                                                                                      |
| by.group    | Logical. Whether to perform pseudo-bulking by group. TRUE by default.                                                                                                            |
| group.id    | Character. The column name of the colData(spe) that contains the group information. Default to 'cell_type'.                                                                      |
| keep.groups | Vector. Values from group.id to include in pseudo- bulking. Default is NULL, where all cells are included in pseudo-bulking.                                                     |
| roi         | Character. The name of the group or cell type on which the roi is computed. If NULL, then no pseudo-bulking will be performed based on roi. Default to NULL.                     |
| roi.only    | Logical. Whether to filter out pseudo-bulk samples formed by cells not in any ROIs. TRUE by default.                                                                             |
| contour     | Character. The name of the group or cell type on which the contour level is computed. If NULL, then no pseudo-bulking will be performed based on contour level. Default to NULL. |

## Value

An edgeR::DGEList object where each library (column) is a pseudo-bulk sample.

Examples

```
data("xenium_bc_spe")

spe <- gridDensity(spe)

coi <- "Breast cancer"

spe <- findROI(spe, coi = coi)

spe <- allocateCells(spe, to.contour=FALSE)

y <- spe2PB(spe, roi = coi)
```

---

|              |                                        |
|--------------|----------------------------------------|
| update_bound | <i>Update the x,y limits of a plot</i> |
|--------------|----------------------------------------|

---

Description

Update the x,y limits of a plot

Usage

```
update_bound(p, x = NULL, y = NULL)
```

Arguments

- |      |                                |
|------|--------------------------------|
| p    | A ggplot() object              |
| x, y | Vectors of new x and y limits. |

Value

a ggplot object

---

|               |                                                   |
|---------------|---------------------------------------------------|
| xenium_bc_spe | <i>Description of the scider example datasets</i> |
|---------------|---------------------------------------------------|

---

Description

- scider-package has 1 datasets:
- xenium\_bc\_spe Example test spatial transcriptomics data in SpatialExperiment format. This test data is randomly subsetting from a publicly available 10X Xenium breast cancer data. Source data: <https://www.10xgenomics.com/resources/datasets/xenium-ffpe-human-breast-with-custom-add-on-panel-1-standard>

**Usage**

```
data("xenium_bc_spe")
```

**Format**

A SpatialExperiment object

**Value**

A SpatialExperiment object

**Examples**

```
data(xenium_bc_spe)
```

---

```
[,SpatialExperiment,ANY,ANY,ANY-method]
```

*Subset for grid level analysis*

---

**Description**

Overwrite the default SpatialExperiment subsetting method to ensure 'grid\_density' is also subsetted if 'gridLevelAnalysis' is TRUE (1 polygon 1 spot)

**Usage**

```
## S4 method for signature 'SpatialExperiment,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]
```

**Arguments**

|      |                                                          |
|------|----------------------------------------------------------|
| x    | A SpatialExperiment object.                              |
| i    | row indices for subsetting.                              |
| j    | col indices for subsetting.                              |
| ...  | further arguments to be passed to or from other methods. |
| drop | passed on to [ indexing operator.                        |

**Value**

A SpatialExperiment object.

# Index

## \* **internal**

scider-package, [3](#)

xenium\_bc\_spe, [45](#)

[, SpatialExperiment, ANY, ANY, ANY-method, [46](#)

allocateCells, [4](#)

BiocNeighborParam, [11](#)

cellsInRegion, [5](#)

cluster\_leiden, [15](#)

cluster\_louvain, [15](#)

colData, [15](#)

computeDensity, [5](#)

computeDensityHex, [6](#)

contour2sf, [7](#)

coord\_hash, [8](#)

corDensity, [8](#)

findKNN, [12](#)

findNbrsGrid, [9](#)

findNbrsSNN, [10](#), [15](#)

findNbrsSpatial, [12](#)

findNeighbors, [12](#)

findROI, [13](#)

getClusters, [14](#)

getContour, [15](#)

getContourRegions, [16](#)

getNiche, [17](#)

globalMoran, [18](#)

grid2df, [19](#)

grid2sf, [20](#)

gridDensity, [20](#)

gridSPE, [22](#)

irlba, [41](#)

localMoran, [23](#), [36](#)

mergeROI, [24](#)

normalizeAssay, [25](#), [41](#)

plotCellCompo, [26](#)

plotContour, [27](#)

plotContourRegion, [28](#)

plotCorHeatmap, [29](#)

plotDensCor, [30](#)

plotDensity, [27](#), [31](#)

plotDR, [32](#)

plotGrid, [31](#), [33](#), [36](#)

plotImage, [27](#), [34](#), [35](#)

plotLISA, [35](#)

plotPCA (plotDR), [32](#)

plotROI, [36](#)

plotSpatial, [27](#), [36](#), [37](#), [38](#)

plotUMAP (plotDR), [32](#)

postSelRegion, [39](#)

realignVisium, [40](#)

realignVisiumHD, [40](#)

reducedDims, [11](#), [15](#), [32](#), [41](#), [42](#)

runPCA, [41](#), [42](#)

runUMAP, [42](#)

scale\_color\_gradientn, [33](#), [38](#)

scale\_fill\_gradientn, [34](#)

scaleFactors, [35](#)

scider (scider-package), [3](#)

scider-package, [3](#)

selectRegion, [43](#)

spe (xenium\_bc\_spe), [45](#)

spe2PB, [44](#)

umap, [42](#)

update\_bound, [45](#)

xenium\_bc\_spe, [45](#)