Package ‘spaSim’

July 2, 2025

Title Spatial point data simulator for tissue images

Version 1.11.0

Date 2023-05-22

Description A suite of functions for simulating spatial patterns of cells in tissue images. Output images are multitype point data in SingleCellExperiment format. Each point represents a cell, with its 2D locations and cell type. Potential cell patterns include background cells, tumour/immune cell clusters, immune rings, and blood/lymphatic vessels.

License Artistic-2.0

URL https://trigosteam.github.io/spaSim/

BugReports https://support.bioconductor.org/t/spaSim

biocViews StatisticalMethod, Spatial, BiomedicalInformatics

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.2

Imports ggplot2, methods, stats, dplyr, spatstat.geom, spatstat.random, SpatialExperiment, SummarizedExperiment, RANN

Depends R (>= 4.2.0)

LazyData true

Suggests RefManageR, BiocStyle, knitr, testthat (>= 3.0.0), sessioninfo, rmarkdown, markdown

Config/testthat/edition 3

VignetteBuilder knitr

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

git_branch devel

git_last_commit 7c4ef82

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-01

Author Yuzhou Feng [aut, cre] (ORCID: <https://orcid.org/0000-0002-2955-4053>), Anna Trigos [aut] (ORCID: <https://orcid.org/0000-0002-5915-2952>)

Maintainer Yuzhou Feng <yuzhou.feng@petermac.org>

Contents

`bg1` *Background cells (4915 cells in a 2000*2000 window)*

Description

A data frame containing 4915 background cells on 2000 by 2000 size window

Usage

```
bg1
```

Format

A data frame with 4915 rows (cells) and 3 columns:

Cell.X.Position	X coordinate
Cell.Y.Position	Y coordinate
Cell.Type	cell type

`format_spe` *format_spe*

Description

Format a data frame object into a `SpatialExperiment` class object where the count assay is empty.

Usage

```
format_spe(df)
```

Arguments

`df`	A data frame where each row contains information about a cell. The columns of the data frame will become the colData of the `SpatialExperiment` object.

Value

An SPE object

`multiple_background_images`

Simulate multiple background images (mixed cell types)

Description

Generate a set of background images with different proportions of mixed cell types all at once. The default values for the arguments give an example of multiple image simulation which enable an automatic multiple image simulation without the specification of any argument.

Usage

```
multiple_background_images(
  bg_sample = bg1,
  idents = c("Tumour", "Immune", "Others"),
  props = list(rep(0.1, 9), seq(0, 0.4, 0.05), seq(0.9, 0.5, -0.05)),
  plot_image = TRUE,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`idents`	String Vector. Names of the cell types to generate.
`props`	List. Each element is a vector of proportions of the corresponding cell type. The length of the vector is how many images to generate. All vectors should be of the same length, also equal to the number of images.
`plot_image`	Boolean. Whether plot the simulated images or not. Default is TRUE.
`plot_colours`	String Vector. If plot_image is TRUE, this param is the corresponding colours for the `idents` arg. Default is NULL, the predefined colour vector would be used for plotting.

Value

A list of SPE objects

See Also

`multiple_images_with_clusters` for simulating multiple images with clusters, and `multiple_images_with_immune_`
for simulating multiple images with immune rings.

Other simulate multiple images functions: `multiple_images_with_clusters()`, `multiple_images_with_immune_ri`

Examples

```
idents <- c("Tumour","Immune","Others")
prop1 <- rep(0.1,9)
prop2 <- seq(0, 0.4, 0.05)
prop3 <- seq(0.9,0.5,-0.05)
set.seed(610)
bg_image_list <- multiple_background_images(bg_sample = bg1,
idents = idents, props = list(prop1, prop2, prop3), plot_image = FALSE)
```

`multiple_images_with_clusters`

Simulate multiple images with clusters

Description

Generate a set of images with different cluster properties. Trying out the default parameters is recommended for understanding this function. The default values for the arguments give an example of multiple image simulation which enable an automatic multiple image simulation without the specification of any argument.

Usage

```
multiple_images_with_clusters(
  bg_sample = bg1,
  cluster_shape = 2,
  prop_infiltration = 0.1,
  cluster_size = seq(200, 1000, 100),
  cluster_loc_x = 0,
  cluster_loc_y = 0,
  plot_image = TRUE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`cluster_shape`	Number. Choose from one of the following pre-designed shapes (1 or 2 for tumour cluster or 3 for immune cluster). The pre-designed shape contains information of the cell names of the cluster, the infiltration cell types, the proportions of infiltration, the cluster size, and the cluster centre locations. In order

	to simulate a set of images, use the arguments below to specify the ranges of the properties. The predefined cell types can not be changed, while users can change them manually after the simulation.
`prop_infiltration`	Numeric Vector. The degree of infiltration. If numeric, all simulated images have the same infiltration degree. If vector, images with a range of different infiltration proportions will be simulated.
`cluster_size`	Numeric Vector. The size of the cluster. If numeric, all simulated images have the same cluster size. If vector, images with a range of different cluster sizes will be simulated. The size should not exceed the limit of the image sides.
`cluster_loc_x`	Numeric or Vector. The X location of the cluster center offset. If numeric, all simulated images have the same center X location. If vector, images with a range of different center locations will be simulated.
`cluster_loc_y`	Numeric or Vector of the same length of `cluster_loc_x`. The Y location of the cluster center offset.
`plot_image`	Boolean Whether plot the simulated images or not.Default is TRUE.
`plot_categories`	String Vector specifying the order of the cell categories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

A list of SPE objects

See Also

`multiple_background_images` for simulating multiple mixed background images, and `multiple_images_with_immun` for simulating multiple images with immune rings.

Other simulate multiple images functions: `multiple_background_images()`, `multiple_images_with_immune_rings`

Examples

```
set.seed(610)
cluster_image_list <- multiple_images_with_clusters(bg_sample=bg1,
cluster_shape=2, prop_infiltration=0.1, cluster_size = seq(200,1000,100),
cluster_loc_x = 0, cluster_loc_y = 0, plot_image = TRUE)
```

`multiple_images_with_immune_rings`

Simulate multiple images with immune rings

Description

Generate a set of images with different immune ring properties. The default values for the arguments give an example of multiple image simulation which enable an automatic multiple image simulation without the specification of any argument.

Usage

```
multiple_images_with_immune_rings(
  bg_sample = bg1,
  cluster_size = 200,
  ring_shape = 1,
  prop_infiltration = 0,
  ring_width = seq(50, 100, 10),
  cluster_loc_x = 0,
  cluster_loc_y = 0,
  prop_ring_infiltration = seq(0, 0.2, 0.05),
  plot_image = TRUE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`cluster_size`	Numeric Vector. The size of the cluster. If numeric, all simulated images have the same cluster size. If vector, images with a range of different cluster sizes will be simulated. The size should not exceed the limit of the image sides.
`ring_shape`	Number. Choose from one of the following pre-designed shapes (1,2 or 3). The pre-designed shape contains information of the cell names of the cluster, the infiltration cell types, the proportions of infiltration, the cluster size, the ring width, the proportions of infiltrated cells into immune rings and the cluster centre locations. In order to simulate a set of images, use the arguments below to specify the ranges of the properties. The predefined cell types can not be changed, while users can change them manually after the simulation.
`prop_infiltration`	Numeric Vector. The degree of infiltration. If numeric, all simulated images have the same infiltration degree. If vector, images with a range of different infiltration proportions will be simulated.
`ring_width`	Numeric Vector. The width of the immune ring. If numeric, all simulated images have the same ring width. If vector, images with a range of different ring widths will be simulated.
`cluster_loc_x`	Numeric or Vector. The X location of the cluster center offset. If numeric, all simulated images have the same center X location. If vector, images with a range of different center locations will be simulated.
`cluster_loc_y`	Numeric or Vector of the same length of `cluster_loc_x`. The Y location of the cluster center offset.
`prop_ring_infiltration`	Numeric or Vector. The degree of tumour infiltration in the region of immune rings.
`plot_image`	Boolean Whether plot the simulated images or not.Default is TRUE.

`plot_categories`	String Vector specifying the order of the cell cateories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

A list of spe objects

See Also

`multiple_background_images` for simulating multiple mixed background images, and `multiple_images_with_clust` for simulating multiple images with clusters.

Other simulate multiple images functions: `multiple_background_images()`, `multiple_images_with_clusters()`

Examples

```
set.seed(610)
ring_image_list <- multiple_images_with_immune_rings(bg_sample = bg1,
ring_shape = 1, prop_infiltration = 0, ring_width = seq(50,100,10),
cluster_size = 200, cluster_loc_x = 0, cluster_loc_y = 0,
prop_ring_infiltration = seq(0, 0.2,0.05), plot_image = TRUE)
```

`plot_cells` *plot_cells*

Description

Produces a scatter plot of the cells in the tissue. Cells are coloured categorically by `Cell.Type` column. Cell categories not specified will be coloured "lightgrey" and labled "Unspecified".

Usage

```
plot_cells(
  spe_object,
  categories_of_interest = NULL,
  colour_vector = NULL,
  feature_colname = "Cell.Type"
)
```

Arguments

`spe_object`	SpatialExperiment object or a data.frame that has cell locations and cell type info.
`categories_of_interest`	String Vector of cell categories to be coloured.
`colour_vector`	String Vector specifying the colours of each cell type.
`feature_colname`	String specifying the column the cell categories belong to.

Value

A plot is returned

`simulate_background_cells`

Simulate background cells

Description

Simulate cell locations. The 2D locations of the cells are simulated and plotted in a rectangular window. Users can specify the window size, cell number and the minimum distance between two cells. All cells have the same cell type, specified by the "Cell.Type" param.

Usage

```
simulate_background_cells(
  n_cells,
  width,
  height,
  method = "Hardcore",
  min_d,
  oversampling_rate = 1.2,
  jitter = 0.3,
  Cell.Type = "Others",
  plot_image = TRUE
)
```

Arguments

`n_cells`	Numeric. Number of cells to simulate in the background.
`width, height`	Numeric. The width and height of the image.
`method`	String. The distribution model for the background cells. Options are "Hardcore" for tumour tissues and "Even" for normal tissues. Default is "Hardcore".
`min_d`	(OPTIONAL) Numeric. Use when `method` is "Hardcore". The minimum distance between two cells.
`oversampling_rate`	(OPTIONAL) Numeric. Use when `method` is "Hardcore". The multiplier for oversampling. Without oversampling, the simulation deletes cells that are within `min_d` from each other, resulting in a less total number of cells than `n_cells`. Default is 1.2 (this should be set based on `n_cells` and `min_d`; should always be larger than 1).
`jitter`	(OPTIONAL) Numeric. Use when `method` is "Even". The uniform distribution parameter to generate the jitter distance for each cell from the vertices of the hexagon.
`Cell.Type`	(OPTIONAL) String. The name of the background cell type. Default is "Others" since there shouldn't be any identity of the background cells.
`plot_image`	(OPTIONAL) Boolean. Default is TRUE.

Details

There are two options for the background cell distribution model. 1) Hardcore model for tumour tissues. This model uses `rHardcore` `spatstat.random`. Since `rHardcore` deletes cells that are within the distance of `min_d` to another cell, resulting in fewer cell specified by users, we here introduce parameter `oversampling_rate` to generate more cells than specified. 2) Normal tissues use an evenly-spaced model where the cells are distributed approximately according to the vertices of a hexagon. The function accomplishes this by generating cells on a hexagonal grid and individually applying a bounded uniform random jitter. In our algorithm, `jitter` is the parameter to define the uniform distribution of the jitter of the cells from the hexagon vertices. If there is a reference image, `jitter` can be estimated by comparing the average minimum distance between cells of the simulated image and the reference image. If without a reference image, We suggest 0.3 as the default value of `jitter` as this gives a sensible outcome.

Value

A data.frame of the simulated background image

See Also

`simulate_mixing` for mixed background simulation, `simulate_clusters` for cluster simulation, `simulate_immune_rings`/`simulate_double_rings` for immune ring simulation, and `simulate_stripes` for vessel simulation.

Other simulate pattern functions: `simulate_clusters()`, `simulate_double_rings()`, `simulate_immune_rings()`, `simulate_mixing()`, `simulate_stripes()`

Examples

```
set.seed(610) # set seed for this background image simulation for reproducibility
background_image <- simulate_background_cells(n_cells = 5000, width = 2000,
                                              height = 2000, method = "Hardcore",
                                              min_d = 10,
                                              oversampling_rate = 1.5,
                                              Cell.Type = "Others",
                                              plot_image = TRUE)
```

`simulate_clusters` *Simulate clusters*

Description

Based on an existing background image, simulate clusters of cells where the same type of cells aggregate. The default values for the arguments give an example of cluster simulation which enable an automatic simulation of clusters without the specification of any argument.

Usage

```
simulate_clusters(
  bg_sample = bg1,
  n_clusters = 2,
  bg_type = "Others",
 cluster_properties = list(C1 = list(name_of_cluster_cell = "Tumour", size = 300, shape
```

```
   = "Oval", centre_loc = data.frame(x = 500, y = 500), infiltration_types =
   c("Immune1", "Others"), infiltration_proportions = c(0.1, 0.05)), C2 =
  list(name_of_cluster_cell = "Immune1", size = 500, shape = "Irregular", centre_loc =
   data.frame(x = 1500, y = 500), infiltration_types = c("Immune2", "Others"),
   infiltration_proportions = c(0.1, 0.05))),
  plot_image = TRUE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	(OPTIONAL) A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`n_clusters`	Numeric. Number of clusters. This must match the `length(cluster_properties)`.
`bg_type`	(OPTIONAL) String. The name of the background cell type if the background sample does not have a "Cell.Type" column. By default is "Others".
`cluster_properties`	List of properties of the clusters. See examples for the format of this arg.
`plot_image`	Boolean. Whether the simulated image is plotted.
`plot_categories`	String Vector specifying the order of the cell categories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

A data.frame of the simulated image

See Also

`simulate_background_cells` for all cell simulation, `simulate_mixing` for mixed background simulation, `simulate_immune_rings`/`simulate_double_rings` for immune ring simulation, and `simulate_stripes` for vessel simulation.

Other simulate pattern functions: `simulate_background_cells()`, `simulate_double_rings()`, `simulate_immune_rings()`, `simulate_mixing()`, `simulate_stripes()`

Examples

```
set.seed(610)
cluster_image <- simulate_clusters(bg_sample = bg1,
n_clusters=2, cluster_properties=list(C1=list(name_of_cluster_cell="Tumour",
size=300, shape="Oval", centre_loc=data.frame("x"=500, "y"=500),
infiltration_types=c("Immune1", "Others"), infiltration_proportions=c(0.1, 0.05)),
C2=list(name_of_cluster_cell="Immune1", size=500, shape="Irregular",
centre_loc=data.frame("x"=1500,"y"=500), infiltration_types=c("Immune2", "Others"),
infiltration_proportions=c(0.1, 0.05))))
```

`simulate_double_rings` *Simulate double immune rings*

Description

Based on an existing background image, simulate double rings of immune cells that surround tumour clusters. The inner ring is the internal margin of a tumour cluster. The outer ring is the external tumour margin. The tumour clusters and the double immune rings are simulated at the same time. The default values for the arguments give an example of double ring simulation which enable an automatic simulation of double rings without the specification of any argument.

Usage

```
simulate_double_rings(
  bg_sample = bg1,
  bg_type = "Others",
  n_dr = 2,
 dr_properties = list(D1 = list(name_of_cluster_cell = "Tumour", size = 300, shape =
    "Circle", centre_loc = data.frame(x = 1000, y = 1000), infiltration_types =
   c("Immune1", "Immune2", "Others"), infiltration_proportions = c(0.15, 0.05, 0.05),
    name_of_ring_cell = "Immune1", immune_ring_width = 150,
   immune_ring_infiltration_types = c("Others"), immune_ring_infiltration_proportions =
    c(0.15), name_of_double_ring_cell = "Immune2", double_ring_width = 100,
   double_ring_infiltration_types = c("Others"), double_ring_infiltration_proportions =
    c(0.15)),
     D2 = list(name_of_cluster_cell = "Tumour", size = 300, shape =
    "Oval", centre_loc = data.frame(x = 1200, y = 1200), infiltration_types =
   c("Immune1", "Immune2", "Others"), infiltration_proportions = c(0.15, 0.05, 0.05),
    name_of_ring_cell = "Immune1", immune_ring_width = 150,
   immune_ring_infiltration_types = c("Others"), immune_ring_infiltration_proportions =
    c(0.15), name_of_double_ring_cell = "Immune2", double_ring_width = 100,
   double_ring_infiltration_types = c("Others"), double_ring_infiltration_proportions =
    c(0.15))),
  plot_image = TRUE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	(OPTIONAL) A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal bg1 background image.
`bg_type`	(OPTIONAL) String The name of the background cell type. By default is "Others".
`n_dr`	Number of double immune rings. This must match the `length(dr_properties)`.
`dr_properties`	List of properties of the double immune rings. Please refer to the examples for the structure of `dr_properties`.

`plot_image`	Boolean. Whether the simulated image is plotted.
`plot_categories`	String Vector specifying the order of the cell categories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

A data.frame of the simulated image

See Also

`simulate_background_cells` for all cell simulation, `simulate_mixing` for mixed background simulation, `simulate_clusters` for cluster simulation, `simulate_immune_rings` for single immune ring simulation, and `simulate_stripes` for vessel simulation.

Other simulate pattern functions: `simulate_background_cells()`, `simulate_clusters()`, `simulate_immune_rings`
`simulate_mixing()`, `simulate_stripes()`

Examples

```
set.seed(610)
# manually define the properties of the immune ring
dr_properties <- list(D1 = list(name_of_cluster_cell = "Tumour",size = 300,
shape = "Circle",centre_loc = data.frame("x" = 1000, "y" = 1000),infiltration_types
= c("Immune1", "Immune2", "Others"),infiltration_proportions = c(0.15, 0.05, 0.05),
name_of_ring_cell = "Immune1",immune_ring_width = 150,immune_ring_infiltration_types
= c("Others"),immune_ring_infiltration_proportions = c(0.15),name_of_double_ring_cell
= "Immune2",double_ring_width = 100,double_ring_infiltration_types = c("Others"),
double_ring_infiltration_proportions = c(0.15)),
D2 = list(name_of_cluster_cell = "Tumour",size = 300,shape = "Oval",centre_loc
= data.frame("x" = 1200, "y" = 1200),infiltration_types = c("Immune1", "Immune2", "Others"),
infiltration_proportions = c(0.15, 0.05, 0.05),name_of_ring_cell = "Immune1",
immune_ring_width = 150,immune_ring_infiltration_types = c("Others"),
immune_ring_infiltration_proportions = c(0.15),name_of_double_ring_cell = "Immune2",
double_ring_width = 100,double_ring_infiltration_types = c("Others"),
double_ring_infiltration_proportions = c(0.15)))

double_ring_image <- simulate_double_rings(bg_sample = bg1,
n_dr = 2, dr_properties = dr_properties)
```

`simulate_immune_rings` *simulate_immune_rings*

Description

Based on an existing background image, simulate rings of immune cells that surround tumour clusters. The tumour clusters and immune rings are simulated at the same time. The default values for the arguments give an example of immune ring simulation which enable an automatic simulation of immune rings without the specification of any argument.

Usage

```
simulate_immune_rings(
  bg_sample = bg1,
  bg_type = "Others",
  n_ir = 2,
 ir_properties = list(I1 = list(name_of_cluster_cell = "Tumour", size = 600, shape =
    "Circle", centre_loc = data.frame(x = 930, y = 1000), infiltration_types =
   c("Immune1", "Immune2", "Others"), infiltration_proportions = c(0.15, 0.05, 0.05),
    name_of_ring_cell = "Immune1", immune_ring_width = 150,
   immune_ring_infiltration_types = c("Others"), immune_ring_infiltration_proportions =
   c(0.15)), I2 = list(name_of_cluster_cell = "Tumour", size = 500, shape = "Oval",
    centre_loc = data.frame(x = 1330, y = 1100), infiltration_types = c("Immune1",

    "Immune2", "Others"), infiltration_proportions = c(0.15, 0.05, 0.05),
    name_of_ring_cell = "Immune1", immune_ring_width = 150,
   immune_ring_infiltration_types = c("Others"), immune_ring_infiltration_proportions =
    c(0.15))),
  plot_image = TRUE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	(OPTIONAL) A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`bg_type`	(OPTIONAL) String The name of the background cell type. By default is "Others".
`n_ir`	Number of immune rings. This must match the arg `length(ir_properties)`.
`ir_properties`	List of properties of the immune rings. Please refer to the examples for the structure of `ir_properties`.
`plot_image`	Boolean. Whether the simulated image is plotted.
`plot_categories`	String Vector specifying the order of the cell categories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

A data.frame of the simulated image

See Also

`simulate_background_cells` for all cell simulation, `simulate_mixing` for mixed background simulation, `simulate_clusters` for cluster simulation, `simulate_double_rings` for double immune ring simulation, and `simulate_stripes` for vessel simulation.

Other simulate pattern functions: `simulate_background_cells()`, `simulate_clusters()`, `simulate_double_rings`
`simulate_mixing()`, `simulate_stripes()`

Examples

```
set.seed(610)
# manually define the properties of the immune ring
ir_properties <- list(I1=list(name_of_cluster_cell="Tumour", size=600,
shape="Circle",centre_loc=data.frame("x"=930, "y"=1000),
infiltration_types=c("Immune1", "Immune2", "Others"), infiltration_proportions
=c(0.15, 0.05, 0.05), name_of_ring_cell="Immune1", immune_ring_width=150,
immune_ring_infiltration_types=c("Others"), immune_ring_infiltration_proportions=c(0.15)),
I2=list(name_of_cluster_cell="Tumour", size=500, shape="Oval",
centre_loc=data.frame("x"=1330, "y"=1100), infiltration_types=c("Immune1", "Immune2", "Others"),
infiltration_proportions=c(0.15, 0.05, 0.05), name_of_ring_cell="Immune1",
immune_ring_width=150, immune_ring_infiltration_types=c("Others"),
immune_ring_infiltration_proportions=c(0.15)))
# simulate immune rings (`n_ir` should match the length of `ir_properties`)
immune_ring_image <- simulate_immune_rings(bg_sample=bg1,
n_ir=2, ir_properties=ir_properties)
```

`simulate_mixing` *Simulate mixed background image*

Description

Based on an existing background image, simulate mixed cell types with specified cell types and proportions. The default values for the arguments give an example of mixed cell type simulation which enable an automatic simulation of mixed cell types without the specification of any argument.

Usage

```
simulate_mixing(
  bg_sample = bg1,
  idents = c("Tumour", "Immune", "Others"),
  props = c(0.2, 0.4, 0.4),
  plot_image = TRUE,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	(OPTIONAL) A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`idents`	String Vector of the mixed cell types.
`props`	Numeric Vector of the proportions of the mixed cell types.
`plot_image`	Boolean. Whether the simulated image is plotted.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `idents` arg. Default is NULL - the predefined colour vector would be used for plotting..

Value

A data.frame of the simulated image

See Also

`simulate_background_cells` for all cell simulation, `simulate_clusters` for cluster simulation, `simulate_immune_rings`/`simulate_double_rings` for immune ring simulation, and `simulate_stripes` for vessel simulation.

Other simulate pattern functions: `simulate_background_cells()`, `simulate_clusters()`, `simulate_double_rings`
`simulate_immune_rings()`, `simulate_stripes()`

Examples

```
set.seed(610)
mix_background <- simulate_mixing(bg_sample=bg1,
idents=c("Tumour","Immune", "Others"), props=c(0.2, 0.4,  0.4),
plot_image=TRUE)
```

`simulate_stripes` *simulate_stripes*

Description

Based on an existing background image, simulate stripes of cells representing vessels. The cell types and widths of the stripes can be specified. The locations of the stripes are randomly simulated. Please refer to the examples to check what properties of the stripes can be specified. The default values for the arguments give an example of vessel simulation which enable an automatic simulation of vessels without the specification of any argument.

Usage

```
simulate_stripes(
  bg_sample = bg1,
  n_stripe_type = 2,
  stripe_properties = list(S1 = list(number_of_stripes = 1, name_of_stripe_cell =
    "Others", width_of_stripe = 80, infiltration_types = c("Immune"),
    infiltration_proportions = c(0.08)), S2 = list(number_of_stripes = 1,
    name_of_stripe_cell = "Others", width_of_stripe = 80, infiltration_types =
    c("Immune"), infiltration_proportions = c(0.08))),
  plot_image = TRUE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	(OPTIONAL) A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords()` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.

`n_stripe_type`	Number of stripe types. Should be the same as `length(stripe_properties`.
`stripe_properties`	List of the properties of the stripes. See examples for the format of the properties. Please refer to the examples for the structure of `stripe_properties`.
`plot_image`	Boolean. Whether the simulated image is plotted. Default is TRUE.
`plot_categories`	String Vector specifying the order of the cell categories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

A data.frame of the simulated image

See Also

`simulate_background_cells` for all cell simulation, `simulate_mixing` for mixed background simulation, `simulate_clusters` for cluster simulation, and `simulate_immune_rings`/`simulate_double_rings` for immune ring simulation

Other simulate pattern functions: `simulate_background_cells()`, `simulate_clusters()`, `simulate_double_rings`
`simulate_immune_rings()`, `simulate_mixing()`

Examples

```
stripe_properties <- list(
S1 = list(
  number_of_stripes = 1,
  name_of_stripe_cell = "Others",
  width_of_stripe = 80,
  infiltration_types = c("Immune"),
  infiltration_proportions = c(0.08)
), S2 = list(
  number_of_stripes = 1,
  name_of_stripe_cell = "Others",
  width_of_stripe = 80,
  infiltration_types = c("Immune"),
  infiltration_proportions = c(0.08)))
set.seed(610)
stripe_image <- simulate_stripes(bg_sample = bg1, n_stripe_type=2,
stripe_properties = stripe_properties, plot_image = TRUE)
```

`TIS` *Tissue Image Simulator (TIS)*

Description

Tissue Image Simulator (TIS) integrates the basic simulation functions in spaSim, including simulating (mixed) background image, clusters, immune rings, double immune rings and stripes. The patterns are simulated on separate layers sequentially (e.g. immune rings are simulated after/on top of background cells). And each layer is also plot sequentially.

Pattern properties (e.g. `properties_of_clusters`) contain the properties of a pattern in the format of list where each element is one pattern. These properties need to be manually defined. Details about the format of the properties see the examples in simulate_clusters simulate_immune_rings simulate_double_rings simulate_stripes

Usage

```
TIS(
  bg_sample = NULL,
  n_cells = NULL,
  width = NULL,
  height = NULL,
  bg_method = NULL,
  min_d = NULL,
  oversampling_rate = 1.2,
  jitter = 0.3,
  names_of_bg_cells = NULL,
  proportions_of_bg_cells = NULL,
  n_clusters = NULL,
  properties_of_clusters = NULL,
  n_immune_rings = NULL,
  properties_of_immune_rings = NULL,
  n_double_rings = NULL,
  properties_of_double_rings = NULL,
  n_stripe_type = NULL,
  properties_of_stripes = NULL,
  image_name = NULL,
  plot_image = FALSE,
  plot_categories = NULL,
  plot_colours = NULL
)
```

Arguments

`bg_sample`	(OPTIONAL) A data frame or `SpatialExperiment` class object with locations of points representing background cells. Further cell types will be simulated based on this background sample. The data.frame or the `spatialCoords` of the SPE object should have colnames including "Cell.X.Positions" and "Cell.Y.Positions". By default use the internal `bg1` background image.
`n_cells`	(OPTIONAL) Number of background cells to simulate. Only when `bg_sample` is NULL.
`width`	(OPTIONAL) Number The width of the image.
`height`	(OPTIONAL) Number The height of the image.
`bg_method`	(OPTIONAL) String specifying the distribution of the background cells. Choose from "Hardcore" and "Even".
`min_d`	(OPTIONAL) Number The minimum distance between two cells.

`oversampling_rate`	(OPTIONAL) Numeric. The multiplier for oversampling. Without oversampling, the simulation deletes cells that are within `min_d` from each other, resulting in a less total number of cells than `n_cells`. Default is 1.2 (this should be set based on `n_cells` and `min_d`; should always be larger than 1).
`jitter`	(OPTIONAL) Numeric. Use when `method` is "Even". The uniform distribution parameter to generate the jitter distance for each cell from the vertices of the hexagon.
`names_of_bg_cells`	(OPTIONAL) Vector The cell types of the background cells. If NULL, the background cells are of one type.
`proportions_of_bg_cells`	(OPTIONAL) Vector The corresponding proportion of each cell type in the background cells.
`n_clusters`	(OPTIONAL) Number of cell clusters. If NULL, no clusters to simulate.
`properties_of_clusters`	(OPTIONAL) List of parameters to define the clusters.
`n_immune_rings`	(OPTIONAL) Number of immune rings. If NULL, no immune rings to simulate.
`properties_of_immune_rings`	(OPTIONAL) List of parameters to define the immune rings.
`n_double_rings`	(OPTIONAL) Number of double immune rings. If NULL, no double rings to simulate.
`properties_of_double_rings`	(OPTIONAL) List of parameters to define the double immune rings.
`n_stripe_type`	(OPTIONAL) Number of stripe (vessel) types. If NULL, no stripes to simulate.
`properties_of_stripes`	(OPTIONAL) List of parameters to define the stripes.
`image_name`	(OPTIONAL) String to name the output tissue image.
`plot_image`	Boolean. Whether the simulated image is plotted.
`plot_categories`	String Vector specifying the order of the cell categories to be plotted. Default is NULL - the cell categories under the "Cell.Type" column would be used for plotting.
`plot_colours`	String Vector specifying the order of the colours that correspond to the `plot_categories` arg. Default is NULL - the predefined colour vector would be used for plotting.

Value

An spe object of the simulated image

Examples

```
set.seed(610)
double_ring_image <- TIS(bg_sample=bg1, n_clusters=1,
properties_of_clusters=list(C1=list( name_of_cluster_cell="Tumour",
size=300, shape="Oval", centre_loc=data.frame("x"=500, "y"=500),
infiltration_types=c("Immune1", "Others"), infiltration_proportions=c(0.1, 0.05))),
plot_image=TRUE)
```

Index