Package 'scSpatialSIM'

December 19, 2023

Title A Point Pattern Simulator for Spatial Cellular Data

Version 0.1.3.3

Description Single cell resolution data has been valuable in learning about tissue microenvironments and interactions between cells or spots. This package allows for the simulation of this level of data, be it single cell or 'spots', in both a univariate (single metric or cell type) and bivariate (2 or more metrics or cell types) ways. As more technologies come to marker, more methods will be developed to derive spatial metrics from the data which will require a way to benchmark methods against each other. Additionally, as the field currently stands, there is not a gold standard method to be compared against. We set out to develop an R package that will allow users to simulate point patterns that can be biologically informed from different tissue domains, holes, and varying degrees of clustering/colocalization. The data can be exported as spatial files and a summary file (like 'HALO'). <https://github.com/FridleyLab/scSpatialSIM/>.

Encoding UTF-8

RoxygenNote 7.2.3

Imports dplyr, ggplot2, magrittr, spatstat.geom, crayon, ggpubr, pbmcapply, spatstat.random, tidyr, utils, methods, proxy

Depends R (>= 4.00)

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0), spatialTIME

VignetteBuilder knitr

License MIT + file LICENSE

Config/testthat/edition 3

URL https://github.com/FridleyLab/scSpatialSIM

NeedsCompilation no

Author Alex Soupir [aut],
Christopher Wilson [aut],
Jordan Creed [aut],
Julia Wrobel [aut],
Oscar Ospina [aut],
Brooke Fridley [aut, cph],
Fridley Lab [cre]

Maintainer Fridley Lab <fridley.lab@moffitt.org>

Repository CRAN

Date/Publication 2023-12-19 18:50:02 UTC

R topics documented:

`CalculateDensity` *Compute Simulation Heatmaps*

Description

Compute Simulation Heatmaps

Usage

```
CalculateDensity(
  sim_object,
  steps = NULL,
  which = "all",
  step_size = 1,
  cores = 1
)
```

Arguments

sim_object	object created with CreateSimulationObject
steps	which simulation steps to compute heatmaps for (Tissue, Holes, Cells, or All)
which	which simulation to compute it for
step_size	resolution of heatmap
cores	number of cpu cores

Value

a new SpatSimObj with probability densities calculated

CreateSimulationObject

Create a spatial simulation object.

Description

This function creates a SpatSimObj for spatial simulations. The object contains information about the simulation window, the number of simulations to perform, and lists of cells, Tissue1/Tissue2, holes, and spatial files.

Usage

```
CreateSimulationObject(window = NULL, sims = NULL, cell_types = 1)
```

Arguments

window	An object of class owin representing the simulation window. If NULL, defaults to a rectangular window of size (0,10) in both x and y directions.
sims	The number of simulations to perform. If NULL or l ess than 1, defaults to 3.
cell_types	The number of cell types. Defaults to 1.

Details

The simulation window is represented by an object of class owin, which specifies the extent and shape of the spatial domain in which the simulations will be performed. If no window is provided, the function creates a rectangular window of size (0,10) in both x and y directions.

The sims argument specifies the number of simulations to perform. If it is set to NULL or less than 1, the function defaults to 3.

The cell_types argument specifies the number of cell types to include in the simulation. By default, the function creates a single cell type, represented by an object of class Cell.

The SpatSimObj is composed of the following classes:

- A Window object of class owin.

- An integer Sims specifying the number of simulations to perform.
- A list of Cells of class Cell.
- A Tissue object of class Tissue1/Tissue2, representing the Tissue1/Tissue2 components of the simulation.
- A Holes object of class Holes, representing holes in the simulation.
- A list of Spatial Files containing any spatial data associated with the simulation.

Value

A SpatSimObj containing the simulation window, the number of simulations to perform, and lists of cells, Tissue1/Tissue2, holes, and spatial files.

Examples

```
CreateSimulationObject()
```

CreateSpatialList *Get Spatial Files from a* SpatSimObj

Description

This function extracts the 'Spatial Files' slot from a Spatial Simulation Object and removes probability columns while converting 'Positive' and 'Negative' in cell assignment columns to 1 and 0, respectively.

Usage

```
CreateSpatialList(sim_object, single_df = FALSE, multihit_action = "random")
```

Arguments

sim_object	A SpatSimObj
single_df	boolean as to whether to collapse the output list of data frames to a single data frame or not. default is FALSE
multihit_action	string of value 'random', 'drop', or 'keep' for cells that are positive for multiple cell assignments

Details

The output of this function creates a list of the spatial files formatted in a way that would allow direct import into a mIF object from the package 'spatialTIME'

Value

A list of data frames, one for each simulated cell type, with cleaned columns

`ExtractParameters` *Get Simulation Parameters*

Description

Get Simulation Parameters

Usage

```
ExtractParameters(sim_object, steps = NULL)
```

Arguments

`sim_object`	simulation object created with `CreateSimulationObject`
`steps`	which parameters to extract from the `SpatSimObj`

Details

This function will return any paramters that are stored in the simulation object. If no simulation steps have been run, then this will return the default parameters. The defaults are over written if new parameters are provided at each step.

Value

a list of S3 clas `SimParams` containing parameters stored in the `sim_object`

Examples

```
#create simulation object
sim_obj = CreateSimulationObject()
#extract default paramters for the Tissue simulation step
defs = ExtractParameters(sim_obj, "Tissue")
```

`GenerateCellPositivity`
Generate Cell Positivity

Description

Generate the probability of a cell being positive given a set of simulation parameters for each file in a SpatSimObj.

Usage

```
GenerateCellPositivity(
  sim_object,
  k = NA,
  xmin = NA,
  xmax = NA,
  ymin = NA,
  ymax = NA,
  sdmin = 1/2,
  sdmax = 2,
  probs = c(0, 1),
  Force = FALSE,
  density_heatmap = FALSE,
  step_size = 1,
  cores = 1,
  shift = 0,
  random = FALSE,
  overwrite = FALSE,
  use_window = FALSE,
  no_kernel = FALSE
)
```

Arguments

sim_object	A SpatSimObj object containing the simulated data.
k	An integer specifying the number of clusters for each simulated patterns
xmin	A numeric value specifying the minimum x value for the kernel.
xmax	A numeric value specifying the maximum x value for the kernel.
ymin	A numeric value specifying the minimum y value for the kernel.
ymax	A numeric value specifying the maximum y value for the kernel.
sdmin	A numeric value specifying the minimum standard deviation for the kernel.
sdmax	A numeric value specifying the maximum standard deviation for the kernel.
probs	Either a vector of c(low probability, high probability) for all cell types or data frame where each row is the low and high probabilities for the cell type. If data frame, number of rows must equal number of cells
Force	A logical value indicating whether to force simulation parameters to be within the simulation window limits.
density_heatmap	A logical value indicating whether to compute a density heatmap for each cell.
step_size	A numeric value specifying the step size for the grid of points within the window.
cores	An integer value specifying the number of cores to use for parallel computation.
shift	A value between 0 and 1 for how related a second or more cell type is to the first
random	whether or not to randomly generate kernels for cells 2 or more, uf TRUE, shift is not used

overwrite	boolean whether to overwrite existing cell kernels and assignments if present
use_window	boolean whether to use the simulation window to set x and y limits
no_kernel	boolean whether to create kernels or to randomly assign points positive based on probs

Details

The function generates the probability of a cell being positive given a set of simulation parameters f or each file in a scSpatialSIM object. It creates a kernel parameter list for k clusters in each simulated pattern and computes the probability of each point in the grid of points within the window for each cell. The function also computes a density heatmap for each cell if density_heatmap is set to TRUE.

Value

Returns the original scSpatialSIM object with additional generated data added to each cell object.

GenerateDistributions *Generate Characteristic Distributions of Cells*

Description

Generate Characteristic Distributions of Cells

Usage

```
GenerateDistributions(
  spatial_data,
  positive_mean = 10,
  negative_mean = 2,
  positive_sd = 2,
  negative_sd = 1
)
```

Arguments

spatial_data	object of either class list or data.frame. Can be created from SpatSimObj with CreateSpatialList
positive_mean, negative_mean	number for mean of which to center the distribution of the positive and negative cell types. Can be single number of vector with length matching number of Cells.
positive_sd, negative_sd	number for the standard deviation around the positive cell type mean. Can be single value or same length as number of Cells.

Value

object with a class that is the same as input `spatial_data` with new columns containing distributions for positive/negative assigned cells

Examples

```
#create simulation object
spatial_data = CreateSimulationObject(sims = 1, cell_types = 1) %>%
  #produce the point pattern
  GenerateSpatialPattern() %>%
  #make tissues
  GenerateTissue(density_heatmap = FALSE, step_size = 0.1, cores = 1) %>%
  #create positive and negative cells
  GenerateCellPositivity(k = 4, sdmin = 3, sdmax = 5,
  density_heatmap = FALSE, step_size = 1, cores = 1, probs = c(0.0, 0.1), shift = 0) %>%
  #convert to a list of spatial data frames
  CreateSpatialList(single_df = FALSE)
spat_data_distribution = GenerateDistributions(spatial_data)
```

GenerateHoles *Generate holes in a spatial simulation object*

Description

This function generates holes (regions of low probability) in a spatial simulation object based on user-defined parameters. The function uses a kernel density estimate to simulate holes, and returns a modified version of the input object with the holes added. The function also has options to compute a density heatmap and to assign points within the holes to be dropped or kept based on a scaled probability value.

Usage

```
GenerateHoles(
  sim_object,
  xmin = NA,
  xmax = NA,
  ymin = NA,
  ymax = NA,
  sdmin = 1/2,
  sdmax = 2,
  hole_prob = c(0.2, 0.35),
  force = FALSE,
  density_heatmap = FALSE,
  step_size = 1,
  cores = 1,
  overwrite = FALSE,
  use_window = FALSE
)
```

Arguments

sim_object	A spatial simulation object of class SpatSimObj
xmin	Minimum x-coordinate for the holes (default: NA)
xmax	Maximum x-coordinate for the holes (default: NA)
ymin	Minimum y-coordinate for the holes (default: NA)
ymax	Maximum y-coordinate for the holes (default: NA)
sdmin	Minimum standard deviation for the kernels (default: 1/2)
sdmax	Maximum standard deviation for the kernels (default: 2)
hole_prob	A vector of length 2 with the minimum and maximum probabilities of a point being within a hole (default: c(0.2, 0.35))
force	Logical; if TRUE, forces the function to simulate outside the window boundaries (default: FALSE)
density_heatmap	Logical; if TRUE, computes a density heatmap (default: FALSE)
step_size	The step size for the grid (default: 1)
cores	The number of cores to use for parallel processing (default: 1)
overwrite	boolean to replace holes if they have been simulated previously
use_window	boolean whether to use the simulation window to set x and y limits

Details

The function first checks that the input object is of the correct class, and that no parameters are NULL. If any parameters are NULL, the function stops with an error message. If the x- and y-ranges for the holes extend beyond the boundaries of the simulation window, the function also stops with an error message, unless the force parameter is set to TRUE. The function then produces kernel parameter lists for each simulated pattern, and generates a grid based on the user-defined step size. If density_heatmap is set to TRUE, the function computes a density heatmap using the CalculateGrid function. Finally, the function computes hole probabilities for each simulated pattern, assigns each point to be dropped or kept based on a scaled probability value, and returns the modified simulation object.

Value

A modified spatial simulation object with holes added

Examples

```
sim_object <- CreateSimulationObject()

#simulate points
sim_object <- GenerateSpatialPattern(sim_object, lambda = 20)

# Generate tissue with default parameters
sim_object <- GenerateTissue(sim_object)
```

```
# Generate holes in the simulation object
sim_object <- GenerateHoles(sim_object, hole_prob = c(0.1, 0.3), force = TRUE)
```

`GenerateSpatialPattern` *Generate Spatial Point Pattern*

Description

Generate a spatial point pattern within the simulation object's window using a Poisson point process.

Usage

```
GenerateSpatialPattern(
  sim_object,
  lambda = 25,
  ...,
  overwrite = FALSE,
  gridded = FALSE,
  grid_shift = 0.5
)
```

Arguments

`sim_object`	A 'SpatSimObj' containing a window.
`lambda`	The intensity of the point pattern Default is 25.
`...`	Additional arguments passed to 'rpoispp'.
`overwrite`	boolean indicating whether or not to replace point patterns if they exist in object
`gridded`	boolean value to whether or not simulate the point pattern in a grid. See details for more.
`grid_shift`	the amount to move alternative columns down when gridded; between -0.5 and 0.5

Details

This function generates a spatial point process within the window of the 'sim_object' using a Poisson point pattern with intensity 'lambda'. The simulated point pattern is added to the 'Patterns' slot of the 'sim_object'. Additional arguments can be passed to the 'rpoispp' function.

The `gridded` parameter is used for simulating point patterns that would represent som spatial transcriptomic technologies such as visium, where rather than like cell being randomly distributed in a sample, the spots where data is measured is evenly spaced.

Value

The updated 'sim_object' with a simulated point process added to the 'Processes' slot.

Examples

```
sim_object <- CreateSimulationObject()
sim_object <- GenerateSpatialPattern(sim_object, lambda = 30)
```

`GenerateTissue` *Generate Tissue*

Description

This function generates a simulated tissue using a specified number of clusters and spatial parameters for each pattern in the simulation object. The tissue is represented by a grid of points with probabilities of belonging to tissue 1 or tissue 2, based on a Gaussian kernel density estimate calculated for each pattern

Usage

```
GenerateTissue(
  sim_object,
  k = NA,
  xmin = NA,
  xmax = NA,
  ymin = NA,
  ymax = NA,
  sdmin = 1/2,
  sdmax = 2,
  force = FALSE,
  density_heatmap = FALSE,
  step_size = 1,
  cores = 1,
  overwrite = FALSE,
  use_window = FALSE
)
```

Arguments

`sim_object`	A `SpatSimObj` created with `CreateSimulationObject`.
`k`	Number of clusters to generate for each pattern
`xmin`	Minimum x-coordinate for cluster centers.
`xmax`	Maximum x-coordinate for cluster centers.
`ymin`	Minimum y-coordinate for cluster centers.
`ymax`	Maximum y-coordinate for cluster centers.
`sdmin`	Minimum standard deviation for cluster kernels.
`sdmax`	Maximum standard deviation for cluster kernels.

force	Logical, whether to force generation of tissue even if the generated cluster centers would fall outside the simulation window. If FALSE, an error will be thrown if cluster centers are outside the window.
density_heatmap	Logical, whether to calculate a density heatmap for the simulated tissue. If TRUE, a grid of points will be generated covering the entire simulation window, and the probability of each grid point belonging to tissue 1 will be calculated based on the generated tissue probability.
step_size	Grid step size for the density heatmap.
cores	Number of cores to use for parallel processing of density calculations.
overwrite	boolean whether to overwrite if tissue kernels already exist
use_window	boolean whether to use the simulation window to set x and y limits

Details

This function generates a simulated tissue for each pattern in the simulation object by first generating k clusters within the specified x and y ranges and with a standard deviation within the specified range. Then, a Gaussian kernel density estimate is calculated for each pattern using the generated clusters as center points and the specified standard deviation as kernel size. The density estimates represent the probability of each point in the simulation window belonging to tissue 1 or tissue 2. If density_heatmap = TRUE, a density heatmap will be calculated using a grid of points covering the entire simulation window. Finally, for each simulated point, the probability of belonging to tissue 1 is calculated based on the kernel density estimate, and the tissue type is assigned with probability proportional to the probability of belonging to tissue 1.

Value

A modified 'Spatial Simulation Object' with updated tissue grids and assigned tissue types for each simulated pattern.

Examples

```
# Create a simulation object with a window and point pattern
sim_object <- CreateSimulationObject()

#simulate points
sim_object <- GenerateSpatialPattern(sim_object, lambda = 20)

# Generate tissue with default parameters
sim_object <- GenerateTissue(sim_object)
```

plot.SpatSimObj *plot function for SpatSimObj*

Description

plot function for SpatSimObj

Usage

```
## S3 method for class 'SpatSimObj'
plot(x, ...)
```

Arguments

x	of class SpatSimObj
...	other things to pass to the plot method for SpatSimObj including nrow, ncol for the number of rows and columns of plots, which pattern to plot, and what which currently only works with "Processes" but may be updated in the future

Value

basic x-y ggplot object

PlotSimulation *Plot Simulation*

Description

Plot different aspects of a SpatSimObj

Usage

```
PlotSimulation(
  sim_object,
  nrow = 1,
  ncol = 1,
  which = 1,
  what = "tissue heatmap"
)
```

Arguments

sim_object	A SpatSimObj
nrow	Number of rows of plots (only applicable when more than one plot is made)
ncol	Number of columns of plots (only applicable when more than one plot is made)
which	Index of the elements of the SpatSimObj to be plotted
what	What to plot ("tissue heatmap", "hole heatmap", or "whole core")

Details

The PlotSimulation function is used to plot different aspects of a SpatSimObj The function takes a sim_object as its first argument, which should be an object of class "Spatial Simulation Object". The function can then be used to plot different aspects of the simulation, such as heatmaps of the tissue or holes, or a plot of the whole core with assigned cells colored by type.

When what is set to "tissue heatmap" or "hole heatmap", the function will plot heatmaps of the specified tissue or hole. When what is set to "whole core", the function will plot the entire core with assigned cells colored by type. Only a single element of the sim_object can be plotted when what is set to "whole core". what equal to "tissue points", "hole points", or "tissue hole points" will result in a point plot of the respective assignments on points.

When more than one plot is made, nrow and ncol can be used to specify the number of rows and columns of the plot grid, respectively.

Value

A plot or a grid of plots, depending on the input arguments

Examples

```
# create a SpatSimObj
sim_object <- CreateSimulationObject()
sim_object = GenerateSpatialPattern(sim_object)
sim_object = GenerateTissue(sim_object, density_heatmap = TRUE, step_size = 1, cores = 1)
# plot a heatmap of tissue 1
PlotSimulation(sim_object, which = 1, what = "tissue heatmap")
```

SummariseSpatial *Summarise Spatial*

Description

Summarise Spatial

Usage

```
SummariseSpatial(spatial_list, markers)
```

Arguments

spatial_list	list of spatial data frames with markers column names
markers	names of columns, probably cell types, that contain 1s and 0s representing positive/negative assignments

Value

data frome with summary counts and proportions for the markers in each spatial data frame

summary.SpatSimObj *summary function for SpatSimObj*

Description

summary function for SpatSimObj

Usage

```
## S3 method for class 'SpatSimObj'
summary(object, ...)
```

Arguments

object	of class SpatSimObj
...	nothing else to pass to summary if object is a SpatSimObj

Value

summary of the SpatSimObj to the terminal

UpdateSimulationWindow *Update the simulation window in a SpatSimObj*

Description

This function updates the simulation window of a SpatSimObj by replacing the existing window with a new one.

Usage

```
UpdateSimulationWindow(sim_object, window = NULL)
```

Arguments

sim_object	A SpatSimObj object
window	A new owin object representing the updated simulation window

Details

The UpdateSimulationWindow() function checks that the input sim_object is of class 'SpatSimObj', that the input window is not null and is of class 'owin'. If these checks pass, the function updates the simulation window in the input sim_object and returns the updated SpatSimObj object.

Value

The updated SpatSimObj object

See Also

CreateSimulationObject

Examples

```
# Create a simulation object
sim_obj <- CreateSimulationObject()

# Update the simulation window
new_window <- spatstat.geom::owin(c(0, 5), c(0, 5))
updated_sim_obj <- UpdateSimulationWindow(sim_obj, window = new_window)
```

---

wm *Round spatstat window*

Description

A mask that turns a window from spatstat round to mimic a window that is similar to most tissue microarray image scans

Usage

```
wm
```

Format

An owin object with many slots describing the size and mask

type mask

Index