

Package ‘sparrpowR’

August 2, 2021

Title Power Analysis to Detect Spatial Relative Risk Clusters

Version 0.2.4

Date 2021-08-02

Maintainer Ian D. Buller <ian.buller@nih.gov>

Description Calculate the statistical power to detect clusters using kernel-based spatial relative risk functions that are estimated using the 'sparr' package. Details about the 'sparr' package methods can be found in the tutorial: Davies et al. (2018) <[doi:10.1002/sim.7577](https://doi.org/10.1002/sim.7577)>. Details about kernel density estimation can be found in J. F. Bithell (1990) <[doi:10.1002/sim.4780090616](https://doi.org/10.1002/sim.4780090616)>. More information about relative risk functions using kernel density estimation can be found in J. F. Bithell (1991) <[doi:10.1002/sim.4780101112](https://doi.org/10.1002/sim.4780101112)>.

License Apache License (>= 2.0)

Encoding UTF-8

RoxygenNote 7.1.1

Depends R (>= 3.5.0), spatstat.geom, spatstat.core, spatstat.linnet, spatstat (>= 2.0-0)

Imports doFuture, doRNG, fields, foreach, future, graphics, grDevices, iterators, lifecycle, raster, sp, sparr, stats

Suggests testthat, spelling, R.rsp

VignetteBuilder R.rsp

Language en-US

URL <https://github.com/machiela-lab/sparrpowR>

BugReports <https://github.com/machiela-lab/sparrpowR/issues>

NeedsCompilation no

Author Ian D. Buller [aut, cre, cph] (<<https://orcid.org/0000-0001-9477-8582>>),
Derek W. Brown [aut, cph] (<<https://orcid.org/0000-0001-8393-1713>>),
Mitchell J. Machiela [ctb] (<<https://orcid.org/0000-0001-6538-9705>>),
Timothy A. Myers [ctb],
NCI [cph, fnd]

Repository CRAN

Date/Publication 2021-08-02 20:20:02 UTC

R topics documented:

sparrpowR-package	2
jitter_power	3
spatial_data	6
spatial_plots	9
spatial_power	11

Index	16
--------------	-----------

sparrpowR-package	<i>The sparrpowR Package: Power Analysis to Detect Spatial Relative Risk Clusters</i>
-------------------	---

Description

Computes the statistical power for the spatial relative risk function.

Details

For a two-group comparison (e.g., cases v. controls) the 'sparrpowR' package calculates the statistical power to detect clusters using the kernel-based spatial relative risk function that is estimated using the 'sparr' package. Details about the 'sparr' package methods can be found in the tutorial: Davies et al. (2018) doi: [10.1002/sim.7577](https://doi.org/10.1002/sim.7577). Details about kernel density estimation can be found in J. F. Bithell (1990) doi: [10.1002/sim.4780090616](https://doi.org/10.1002/sim.4780090616). More information about relative risk functions using kernel density estimation can be found in J. F. Bithell (1991) doi: [10.1002/sim.4780101112](https://doi.org/10.1002/sim.4780101112).

This package provides a function to compute the statistical power for the spatial relative risk function with various theoretical spatial sampling strategies. The 'sparrpowR' package also provides a function to compute the statistical power for the spatial relative risk function for scenarios where one group (e.g., cases) have been observed and a theoretical sampling strategy for the second group (e.g., controls) is desired. The 'sparrpowR' package also provides visualization of data and statistical power.

Key content of the 'sparrpowR' package include:

Theoretical Spatial Sampling

[spatial_data](#) Generates random two-group data for a spatial relative risk function.

Statistical Power

[spatial_power](#) Computes the statistical power of a spatial relative risk function using randomly generated data.

[jitter_power](#) Computes the statistical power of a spatial relative risk function using previously collected data.

Data Visualization

[spatial_plots](#) Visualizes multiple plots of output from [spatial_data](#), [spatial_power](#) and [jitter_power](#) functions.

Dependencies

The 'sparrpowR' package relies heavily upon `sparr`, `spatstat.core`, `spatstat.geom`, and `raster` for computing the statistical power and visualizing the output. Computation can be performed in parallel using `doFuture`, `multisession`, `doRNG`, and `foreach`. Basic visualizations rely on the `plot.ppp` and `image.plot` functions.

Author(s)

Ian D. Buller

Occupational and Environmental Epidemiology Branch, Division of Cancer Epidemiology and Genetics, National Cancer Institute, National Institutes of Health, Rockville, Maryland, USA.

Derek W. Brown

Integrative Tumor Epidemiology Branch, Division of Cancer Epidemiology and Genetics, National Cancer Institute, National Institutes of Health, Rockville, Maryland, USA.

Maintainer: I.D.B. <ian.buller@nih.gov>

jitter_power

Power of SRR function for previously collected data.

Description

Compute the statistical power of a spatial relative risk function using previously collected data.

Usage

```
jitter_power(
  obs_data,
  sim_total = 2,
  samp_control = c("uniform", "CSR", "MVN"),
  s_control = 1,
  alpha = 0.05,
  p_correct = "none",
  parallel = FALSE,
  n_core = 2,
  verbose = TRUE,
  ...,
  cascon = lifecycle::deprecated(),
  lower_tail = lifecycle::deprecated(),
  upper_tail = lifecycle::deprecated()
)
```

Arguments

`obs_data` A bivariate point pattern (a multitype point pattern of object of class "ppp") with two types of points in a factor valued mark.

sim_total	Integer, specifying the number of simulation iterations to perform.
samp_control	Character string specifying whether to randomize the control locations uniformly (samp_control="uniform"), with complete spatial randomness (samp_control="CSR"), or multivariate normal (samp_control="MVN").
s_control	Optional. Numeric value for the standard deviation of the multivariate normal distribution in the units of the obs_data. The default value (1) assumes a unit square window. Ignored if Ignored if samp_control="uniform" or samp_control="CSR".
alpha	Optional. Numeric value of the critical p-value (default=0.05).
p_correct	Optional. Character string specifying whether to apply a correction for multiple comparisons including a False Discovery Rate p_correct = "FDR", a Sidak correction p_correct = "uncorrelated Sidak", and a Bonferroni correction p_correct = "uncorrelated Bonferroni". If p_correct = "none" (the default), then no correction is applied.
parallel	Logical. If TRUE, will execute the function in parallel. If FALSE (the default), will not execute the function in parallel.
n_core	Optional. Integer specifying the number of CPU cores on current host to use for parallelization (the default is 2 cores).
verbose	Logical. If TRUE (the default), will print function progress during execution. If FALSE, will not print.
...	Arguments passed to risk to select bandwidth, edge correction, and resolution.
cascon	[Deprecated] cascon is no longer supported and this function will output power for case-only and case/control clustering. This argument has been moved to spatial_plots function.
lower_tail	[Deprecated] lower_tail is no longer supported; this function uses alpha to set the critical p-value.
upper_tail	[Deprecated] upper_tail is no longer supported; this function uses alpha to set the critical p-value.

Details

This function computes the statistical power of the spatial relative risk function (nonparametric estimate of relative risk by kernel smoothing) for previously collected studies with known case and control locations.

The function uses the [risk](#) function to estimate the spatial relative risk function and forces the tolerate argument to be TRUE in order to calculate asymptotic p-values.

If samp_control = "uniform" the control locations are randomly generated uniformly within the dow of obs_data. By default, the resolution is an integer value of 128 and can be specified using the resolution argument in the internally called [risk](#) function.

If samp_control = "CSR" the control locations are randomly generated assuming complete spatial randomness (homogeneous Poisson process) within the dow of obs_data with a lambda = number of controls / [resolution x resolution]. By default, the resolution is an integer value of 128 and can be specified using the resolution argument in the internally called [risk](#) function.

If `samp_control = "MVN"` the control locations are randomly generated assuming a multivariate normal distribution *centered at each observed location*. The optional argument `s_control` specifies the standard deviation of the multivariate normal distribution (1 by default) in the units of the `obs_data`.

The function computes a one-sided hypothesis test for case clustering ($\alpha = 0.05$ by default). The function also computes a two-sided hypothesis test for case clustering and control clustering (lower tail = 0.025 and upper tail = 0.975).

The function has functionality for a correction for multiple testing. If `p_correct = "FDR"`, calculates a False Discovery Rate by Benjamini and Hochberg. If `p_correct = "Sidak"`, calculates a Sidak correction. If `p_correct = "Bonferroni"`, calculates a Bonferroni correction. If `p_correct = "none"` (the default), then the function does not account for multiple testing and uses the uncorrected alpha level. See the internal `pval_correct` function documentation for more details.

Value

An object of class "list". This is a named list with the following components:

`sim` An object of class 'rrs' for the first iteration of simulated data.

`out` An object of class 'rrs' for the observed spatial relative risk function without randomization.

`rr_mean` Vector of length [resolution x resolution] of the mean relative risk values at each gridded knot.

`pval_mean` Vector of length [resolution x resolution] of the mean asymptotic p-value at each gridded knot.

`rr_sd` Vector of length [resolution x resolution] of the standard deviation of relative risk values at each gridded knot.

`pval_prop_cascon` Vector of length [resolution x resolution] of the proportion of asymptotic p-values that were significant for both case and control locations at each gridded knot.

`pval_prop_cas` Vector of length [resolution x resolution] of the proportion of asymptotic p-values that were significant for only case locations at each gridded knot.

`rx` Vector of length [resolution x resolution] of the x-coordinates of each gridded knot.

`ry` Vector of length [resolution x resolution] of the y-coordinates of each gridded knot.

`n_cas` Vector of length `sim_total` of the number of case locations simulated in each iteration.

`n_con` Vector of length `sim_total` of the number of control locations simulated in each iteration.

`bandw` Vector of length `sim_total` of the bandwidth (of numerator) used in each iteration.

`s_obs` Vector of length `sim_total` of the global s statistic.

`t_obs` Vector of length `sim_total` of the global t statistic.

`alpha` Vector of length `sim_total` of the (un)corrected critical p-values.

See Also

[risk](#) for additional arguments for bandwidth selection, edge correction, and resolution.

Examples

```
# Using the 'chorley' data set from 'spatstat.data' package
data(chorley)
f1 <- jitter_power(obs_data = unique(chorley),
                  samp_control = "CSR",
                  verbose = FALSE)
```

spatial_data

Simulate random data for SRR function

Description

Generate random two-group data for a spatial relative risk function.

Usage

```
spatial_data(
  win = spatstat.geom::unit.square(),
  sim_total = 2,
  x_case,
  y_case,
  samp_case = c("uniform", "MVN", "CSR", "IPP"),
  samp_control = c("uniform", "systematic", "MVN", "CSR", "IPP", "clustered"),
  x_control = NULL,
  y_control = NULL,
  n_case = NULL,
  n_control = NULL,
  npc_control = NULL,
  r_case = NULL,
  r_control = NULL,
  s_case = NULL,
  s_control = NULL,
  l_case = NULL,
  l_control = NULL,
  e_control = NULL,
  ...
)
```

Arguments

win	Window in which to simulate the random data. An object of class "owin" or something acceptable to <code>as.owin</code> .
sim_total	Integer, specifying the number of simulation iterations to perform.
x_case	Numeric value, or numeric vector, of x-coordinate(s) of case cluster(s).
y_case	Numeric value, or numeric vector, of y-coordinate(s) of case cluster(s).

samp_case	Character string specifying whether to randomize the case locations uniformly (samp_control = "uniform"), multivariate normal (samp_control = "MVN"), with complete spatial randomness (samp_control = "CSR"), or using the inhomogeneous Poisson process (samp_control = "IPP") around each case centroid.
samp_control	Character string specifying whether to randomize the control locations uniformly (samp_control = "uniform"), systematically (samp_control = "systematic"), multivariate normal (samp_control = "MVN"), with complete spatial randomness (samp_control = "CSR"), using the inhomogeneous Poisson process (samp_control = "IPP"), or a realization of the Neyman-Scott cluster process (samp_control = "clustered").
x_control	Numeric value, or numeric vector, of x-coordinate(s) of case cluster(s). Ignored if samp_control != "MVN".
y_control	Numeric value, or numeric vector, of y-coordinate(s) of case cluster(s). Ignored if samp_control != "MVN".
n_case	Numeric value, or numeric vector, of the sample size for case locations in each cluster.
n_control	Numeric value, or numeric vector, of the sample size for control locations in each cluster.
npc_control	Optional. Numeric value of the number of clusters of control locations. Ignored if samp_control != "clustered".
r_case	Optional. Numeric value, or numeric vector, of radius (radii) of case cluster(s) in the units of win. Ignored if samp_case = "MVN".
r_control	Optional. Numeric value, or numeric vector, of radius (radii) of control cluster(s) in the units of win. Ignored if samp_control != "clustered".
s_case	Optional. Numeric value, or numeric vector, for the standard deviation(s) of the multivariate normal distribution for case locations in the units of win. Ignored if samp_control != "MVN".
s_control	Optional. Numeric value, or numeric vector, for the standard deviation(s) of the multivariate normal distribution for control locations in the units of win. Ignored if samp_control != "MVN".
l_case	Optional. A single positive number, a vector of positive numbers, a function(x,y, ...), or a pixel image. Intensity of the Poisson process for case clusters. Ignored if samp_control != "IPP".
l_control	Optional. A single positive number, a vector of positive numbers, a function(x,y, ...), or a pixel image. Intensity of the Poisson process for control clusters. Ignored if samp_control = "uniform", samp_control = "systematic", samp_control = "MVN", or samp_control = "CSR".
e_control	Optional. A single non-negative number for the size of the expansion of the simulation window for generating parent points. Ignored if samp_control != "clustered".
...	Arguments passed to runifdisc , disc , rpoispp , rsyst , or rNeymanScott depending on samp_control or samp_control.

Details

This function generates random data for a spatial relative risk function (nonparametric estimate of relative risk by kernel smoothing) using various random point pattern generators from the [spatstat.core](#) package to generate data.

If `samp_case = "uniform"` the case locations are randomly generated uniformly within a disc of radius `r_case` (or discs of radii `r_case`) centered at coordinates `(x_case, y_case)`.

If `samp_case = "MVN"` the case locations are randomly generated assuming a multivariate normal distribution centered at coordinates `(x_case, y_case)` with a standard deviation of `s_case`.

If `samp_case = "CSR"` the case locations are randomly generated assuming complete spatial randomness (homogeneous Poisson process) within a disc of radius `r_case` (or discs of radii `r_case`) centered at coordinates `(x_case, y_case)` with $\lambda = n_case / \text{area of disc}$.

If `samp_case = "IPP"` the case locations are randomly generated assuming an inhomogeneous Poisson process with a disc of radius `r_case` (or discs of radii `r_case`) centered at coordinates `(x_case, y_case)` with $\lambda = l_case$, a function.

If `samp_control = "uniform"` the control locations are randomly generated uniformly within the window `win`.

If `samp_control = "systematic"` the control locations are randomly generated systematically within the window `win` consisting of a grid of equally-spaced points with a random common displacement.

If `samp_control = "MVN"` the control locations are randomly generated assuming a multivariate normal distribution centered at coordinates `(x_control, y_control)` with a standard deviation of `s_control`.

If `samp_control = "CSR"` the control locations are randomly generated assuming complete spatial randomness (homogeneous Poisson process) within the window `win` with a $\lambda = n_control / [\text{resolution} \times \text{resolution}]$. By default, the resolution is an integer value of 128 and can be specified using the `resolution` argument in the internally called [risk](#) function.

If `samp_control = "IPP"` the control locations are randomly generated assuming an inhomogeneous Poisson process within the window `win` with a $\lambda = l_control$, a function.

If `samp_control = "clustered"` the control locations are randomly generated with a realization of the Neyman-Scott process within the window `win` with the intensity of the Poisson process cluster centres ($\kappa = l_control$), the size of the expansion of the simulation window for generative parent points (`e_control`), and the radius (or radii) of the disc for each cluster (`r_control`).

Value

An object of class "ppplist". This is a list of marked point patterns that have a single mark with two levels: case and control.

See Also

[runifdisc](#), [disc](#), [rpoispp](#), [rsyst](#), or [rNeymanScott](#) for additional arguments for random point pattern generation.

Examples

```

spatial_data(x_case = c(0.25, 0.5, 0.75),
             y_case = c(0.75, 0.25, 0.75),
             samp_case = "MVN",
             samp_control = "MVN",
             x_control = c(0.25, 0.5, 0.75),
             y_control = c(0.75, 0.25, 0.75),
             n_case = 100,
             n_control = c(100,500,300),
             s_case = c(0.05,0.01,0.05),
             s_control = 0.05,
             verbose = FALSE)

```

spatial_plots

Plots for statistical power estimates of the SRR function

Description

Create multiple plots of output from [spatial_data](#), [spatial_power](#) and [jitter_power](#) functions.

Usage

```

spatial_plots(
  input,
  p_thresh = 0.8,
  cascon = FALSE,
  n_sim = 1,
  cols = c("#000000", "#CCCCCC", "#FF0000", "#0000FF"),
  chars = c(1, 1),
  sizes = c(1, 1),
  scale = 1,
  plot_pts = TRUE,
  plot_title = TRUE,
  plot_text = FALSE,
  plot_axes = FALSE,
  plot_square = FALSE,
  horizontal = TRUE,
  ...
)

```

Arguments

input	An object of class "ppplist" from the spatial_data function or an object of class "list" from the spatial_power or jitter_power functions.
p_thresh	A numeric value between 0 and 1 (default = 0.8) for the power threshold.

cascon	Logical. If TRUE, displays the statistical power to detect case clusters and control clusters (two-tailed hypothesis). If FALSE (the default), displays the statistical power to detect case clusters only (one-tailed, lower-tail hypothesis).
n_sim	Integer. The number of simulated iterations to plot. The default is one (1).
cols	Character string of length four (4) specifying the colors for plotting: 1) insufficiently powered, 2) sufficiently powered, 3) case locations, 4) control locations. The default colors in hex are <code>c("#000000", "#CCCCCC", "#FF0000", "#0000FF")</code> or <code>c("grey0", "grey80", "red", "blue")</code> .
chars	Vector of integers or character string of length two (2) for symbols of case and control locations. Default is <code>c(1, 1)</code> .
sizes	Vector of integers of length two (2) for the size of the symbols for case and control locations. Default is <code>c(1, 1)</code> .
scale	Integer. A graphical expansion factor (default is 1) for text (and point) size within plots. Intended for scaling plot features with plot resolution.
plot_pts	Logical. If TRUE (the default), the points from the first simulation iteration will be added to second plot. Not if FALSE.
plot_title	Logical. If TRUE (the default), a title will be included in the plot(s). Not if FALSE.
plot_text	Logical. If TRUE, the local statistical power will be printed at each grid cell. Not if FALSE (the default).
plot_axes	Logical. If TRUE, the axes with labels will be included in the plot(s). Not if FALSE (the default).
plot_square	Logical. If TRUE, the plot will have margins with similar units. Not if FALSE (the default).
horizontal	Logical. If TRUE (the default), the color key will be displayed horizontally, below the plots. If FALSE, the color key will be displayed vertically, to the right of the plots.
...	Arguments passed to <code>plot.ppp</code> and <code>image.plot</code> for additional graphical features.

Value

This function produces up to three plots: 1) example input, 2) local power, and 3) local power above a threshold if the input is from the `spatial_power` or `jitter_power` functions. If the input is from the `spatial_data` function, this function will only display the first plot.

Examples

```
# run spatial_power(), jitter_power(), or spatial_data()
sim_power <- spatial_power(x_case = c(0.25, 0.5, 0.75),
  y_case = c(0.75, 0.25, 0.75),
  samp_case = "MVN",
  samp_control = "MVN",
  x_control = c(0.25, 0.5, 0.75),
  y_control = c(0.75, 0.25, 0.75),
```

```

n_case = 100,
n_control = c(100,500,300),
s_case = c(0.05,0.01,0.05),
s_control = 0.05,
verbose = FALSE)

# run spatial_plots()
spatial_plots(input = sim_power)

```

spatial_power

Power of SRR function for randomly generated data.

Description

Compute the statistical power of a spatial relative risk function using randomly generated data.

Usage

```

spatial_power(
  win = spatstat.geom::unit.square(),
  sim_total = 2,
  x_case,
  y_case,
  samp_case = c("uniform", "MVN", "CSR", "IPP"),
  samp_control = c("uniform", "systematic", "MVN", "CSR", "IPP", "clustered"),
  x_control = NULL,
  y_control = NULL,
  n_case = NULL,
  n_control = NULL,
  npc_control = NULL,
  r_case = NULL,
  r_control = NULL,
  s_case = NULL,
  s_control = NULL,
  l_case = NULL,
  l_control = NULL,
  e_control = NULL,
  alpha = 0.05,
  p_correct = "none",
  verbose = TRUE,
  parallel = FALSE,
  n_core = 2,
  ...,
  cascon = lifecycle::deprecated(),
  lower_tail = lifecycle::deprecated(),
  upper_tail = lifecycle::deprecated()
)

```

Arguments

win	Window in which to simulate the random data. An object of class "owin" or something acceptable to <code>as.owin</code> .
sim_total	Integer, specifying the number of simulation iterations to perform.
x_case	Numeric value, or numeric vector, of x-coordinate(s) of case cluster(s).
y_case	Numeric value, or numeric vector, of y-coordinate(s) of case cluster(s).
samp_case	Character string specifying whether to randomize the case locations uniformly (<code>samp_control="uniform"</code>), multivariate normal (<code>samp_control="MVN"</code>), with complete spatial randomness (<code>samp_control="CSR"</code>), or using the inhomogeneous Poisson process (<code>samp_control="IPP"</code>) around each case centroid.
samp_control	Character string specifying whether to randomize the control locations uniformly (<code>samp_control="uniform"</code>), systematically (<code>samp_control="systematic"</code>), multivariate normal (<code>samp_control="MVN"</code>), with complete spatial randomness (<code>samp_control="CSR"</code>), using the inhomogeneous Poisson process (<code>samp_control="IPP"</code>), or a realization of the Neyman-Scott cluster process (<code>samp_control="clustered"</code>).
x_control	Numeric value, or numeric vector, of x-coordinate(s) of case cluster(s). Ignored if <code>samp_control!="MVN"</code> .
y_control	Numeric value, or numeric vector, of y-coordinate(s) of case cluster(s). Ignored if <code>samp_control!="MVN"</code> .
n_case	Numeric value, or numeric vector, of the sample size for case locations in each cluster.
n_control	Numeric value, or numeric vector, of the sample size for control locations in each cluster.
npc_control	Optional. Numeric value of the number of clusters of control locations. Ignored if <code>samp_control!="clustered"</code> .
r_case	Optional. Numeric value, or numeric vector, of radius (radii) of case cluster(s) in the units of win. Ignored if <code>samp_case="MVN"</code> .
r_control	Optional. Numeric value, or numeric vector, of radius (radii) of control cluster(s) in the units of win. Ignored if <code>samp_control!="clustered"</code> .
s_case	Optional. Numeric value, or numeric vector, for the standard deviation(s) of the multivariate normal distribution for case locations in the units of win. Ignored if <code>samp_control!="MVN"</code> .
s_control	Optional. Numeric value, or numeric vector, for the standard deviation(s) of the multivariate normal distribution for control locations in the units of win. Ignored if <code>samp_control!="MVN"</code> .
l_case	Optional. A single positive number, a vector of positive numbers, a function(x,y,...), or a pixel image. Intensity of the Poisson process for case clusters. Ignored if <code>samp_control!="IPP"</code> .
l_control	Optional. A single positive number, a vector of positive numbers, a function(x,y,...), or a pixel image. Intensity of the Poisson process for control clusters. Ignored if <code>samp_control="uniform"</code> , <code>samp_control="systematic"</code> , <code>samp_control="MVN"</code> , or <code>samp_control="CSR"</code> .

e_control	Optional. A single non-negative number for the size of the expansion of the simulation window for generating parent points. Ignored if samp_control!="clustered".
alpha	Optional. Numeric value of the critical p-value (default=0.05).
p_correct	Optional. Character string specifying whether to apply a correction for multiple comparisons including a False Discovery Rate p_correct = "FDR", a Sidak correction p_correct = "uncorrelated Sidak", and a Bonferroni correction p_correct = "uncorrelated Bonferroni". If p_correct = "none" (the default), then no correction is applied.
verbose	Logical. If TRUE (the default), will print function progress during execution. If FALSE, will not print.
parallel	Logical. If TRUE, will execute the function in parallel. If FALSE (the default), will not execute the function in parallel.
n_core	Optional. Integer specifying the number of CPU cores on current host to use for parallelization (the default is 2 cores).
...	Arguments passed to <code>runifdisc</code> , <code>disc</code> , <code>rpoispp</code> , <code>rsyst</code> , or <code>rNeymanScott</code> depending on <code>samp_control</code> or <code>samp_control</code> . Arguments also passed to <code>risk</code> to select bandwidth, edge correction, and resolution.
cascon	[Deprecated] cascon is no longer supported and this function will output power for case-only and case/control clustering. This argument has been moved to <code>spatial_plots</code> function.
lower_tail	[Deprecated] lower_tail is no longer supported; this function uses alpha to set the critical p-value.
upper_tail	[Deprecated] upper_tail is no longer supported; this function uses alpha to set the critical p-value.

Details

This function computes the statistical power of the spatial relative risk function (nonparametric estimate of relative risk by kernel smoothing) for randomly generated data using various random point pattern generators from the `spatstat.core` package.

The function uses the `risk` function to estimate the spatial relative risk function and forces the `tolerate` argument to be TRUE in order to calculate asymptotic p-values.

If `samp_case = "uniform"` the case locations are randomly generated uniformly within a disc of radius `r_case` (or discs of radii `r_case`) centered at coordinates `(x_case, y_case)`.

If `samp_case = "MVN"` the case locations are randomly generated assuming a multivariate normal distribution centered at coordinates `(x_case, y_case)` with a standard deviation of `s_case`.

If `samp_case = "CSR"` the case locations are randomly generated assuming complete spatial randomness (homogeneous Poisson process) within a disc of radius `r_case` (or discs of radii `r_case`) centered at coordinates `(x_case, y_case)` with $\lambda = n_case / \text{area of disc}$.

If `samp_case = "IPP"` the case locations are randomly generated assuming an inhomogeneous Poisson process with a disc of radius `r_case` (or discs of radii `r_case`) centered at coordinates `(x_case, y_case)` with $\lambda = l_case$, a function.

If `samp_control = "uniform"` the control locations are randomly generated uniformly within the window `win`.

If `samp_control = "systematic"` the control locations are randomly generated systematically within the window `win` consisting of a grid of equally-spaced points with a random common displacement.

If `samp_control = "MVN"` the control locations are randomly generated assuming a multivariate normal distribution centered at coordinates `(x_control, y_control)` with a standard deviation of `s_control`.

If `samp_control = "CSR"` the control locations are randomly generated assuming complete spatial randomness (homogeneous Poisson process) within the window `win` with a `lambda = n_control / [resolution x resolution]` By default, the resolution is an integer value of 128 and can be specified using the `resolution` argument in the internally called `risk` function.

If `samp_control = "IPP"` the control locations are randomly generated assuming an inhomogeneous Poisson process within the window `win` with a `lambda = l_control`, a function.

If `samp_control = "clustered"` the control locations are randomly generated with a realization of the Neyman-Scott process within the window `win` with the intensity of the Poisson process cluster centres (`kappa = l_control`), the size of the expansion of the simulation window for generative parent points (`e_control`), and the radius (or radii) of the disc for each cluster (`r_control`).

The function computes a one-sided hypothesis test for case clustering (`alpha = 0.05` by default). The function also computes a two-sided hypothesis test for case clustering and control clustering (lower tail = 0.025 and upper tail = 0.975).

The function has functionality for a correction for multiple testing. If `p_correct = "FDR"`, calculates a False Discovery Rate by Benjamini and Hochberg. If `p_correct = "Sidak"`, calculates a Sidak correction. If `p_correct = "Bonferroni"`, calculates a Bonferroni correction. If `p_correct = "none"` (the default), then the function does not account for multiple testing and uses the uncorrected alpha level. See the internal `pval_correct` function documentation for more details.

Value

An object of class "list". This is a named list with the following components:

`sim` An object of class 'rrs' for the first iteration of simulated data.

`out` An object of class 'rrs' for the observed spatial relative risk function without randomization.

`rr_mean` Vector of length `[resolution x resolution]` of the mean relative risk values at each gridded knot.

`pval_mean` Vector of length `[resolution x resolution]` of the mean asymptotic p-value at each gridded knot.

`rr_sd` Vector of length `[resolution x resolution]` of the standard deviation of relative risk values at each gridded knot.

`pval_prop_cascon` Vector of length `[resolution x resolution]` of the proportion of asymptotic p-values that were significant for both case and control locations at each gridded knot.

`pval_prop_cas` Vector of length `[resolution x resolution]` of the proportion of asymptotic p-values that were significant for only case locations at each gridded knot.

`rx` Vector of length `[resolution x resolution]` of the x-coordinates of each gridded knot.

`ry` Vector of length `[resolution x resolution]` of the y-coordinates of each gridded knot.

`n_cas` Vector of length `sim_total` of the number of case locations simulated in each iteration.

`n_con` Vector of length `sim_total` of the number of control locations simulated in each iteration.

bandw Vector of length *sim_total* of the bandwidth (of numerator) used in each iteration.
s_obs Vector of length *sim_total* of the global s statistic.
t_obs Vector of length *sim_total* of the global t statistic.
alpha Vector of length *sim_total* of the (un)corrected critical p-values.

Examples

```
spatial_power(x_case = c(0.25, 0.5, 0.75),  
              y_case = c(0.75, 0.25, 0.75),  
              samp_case = "MVN",  
              samp_control = "MVN",  
              x_control = c(0.25, 0.5, 0.75),  
              y_control = c(0.75, 0.25, 0.75),  
              n_case = 100,  
              n_control = c(100, 500, 300),  
              s_case = c(0.05, 0.01, 0.05),  
              s_control = 0.05,  
              verbose = FALSE)
```

Index

* package

sparrpowR-package, 2

as.owin, 6, 12

disc, 7, 8, 13

doFuture, 3

doRNG, 3

foreach, 3

image.plot, 3, 10

jitter_power, 2, 3, 9, 10

multisession, 3

plot.ppp, 3, 10

raster, 3

risk, 4, 5, 8, 13, 14

rNeymanScott, 7, 8, 13

rpoispp, 7, 8, 13

rsyst, 7, 8, 13

runifdisc, 7, 8, 13

sparr, 3

sparrpowR (sparrpowR-package), 2

sparrpowR-package, 2

spatial_data, 2, 6, 9, 10

spatial_plots, 2, 9

spatial_power, 2, 9, 10, 11

spatstat.core, 3, 8, 13

spatstat.geom, 3