

# Package ‘sparrpowR’

June 17, 2020

**Title** Power Analysis to Detect Spatial Relative Clusters

**Version** 0.1.3

**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>. Details about kernel density estimation can be found in J. F. Bethell (1990) <doi: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>.

**License** Apache License (>= 2.0)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.0.2

**Depends** R (>= 3.5.0), spatstat

**Imports** doParallel, fields, foreach, graphics, grDevices, parallel, raster, sp, sparr, stats, utils

**Suggests** tidyverse, testthat, spelling, R.rsp

**VignetteBuilder** R.rsp

**Language** en-US

**NeedsCompilation** no

**Author** Ian D. Buller [aut, cre, cph],  
Derek W. Brown [aut, cph],  
Mitchell J. Machiela [ctb],  
Timothy A. Myers [ctb],  
NCI [cph, fnd]

**Repository** CRAN

**Date/Publication** 2020-06-16 23:00:02 UTC

## R topics documented:

jitter_power	2
spatial_data	4
spatial_plots	7
spatial_power	8

<b>Index</b>	<b>13</b>
--------------	-----------

---

jitter_power	<i>Power of SRR function for previously collected data.</i>
--------------	---

---

### Description

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

### Usage

```
jitter_power(
  obs_data,
  sim_total,
  samp_control = c("uniform", "CSR", "MVN"),
  s_control = 1,
  cascon = FALSE,
  lower_tail = 0.025,
  upper_tail = 0.975,
  parallel = FALSE,
  n_core = 2,
  verbose = TRUE,
  ...
)
```

### 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".
cascon	Logical. If FALSE (the default) computes the power to detect only relative case clustering. If TRUE, computes power to detect both case and control clustering.
lower_tail	Optional. Numeric value of lower p-value threshold (default=0.025).

upper_tail	Optional. Numeric value of upper p-value threshold (default=0.975). Ignored if cascon=FALSE.
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 <a href="#">risk</a> to select bandwidth, edge correction, and resolution.

### 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 window 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 window 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.

### 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.

rr\_mean Vector of length [resolution x resolution] of the proportion of asymptotic p-values that were significant 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.

`rx` 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.

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

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

### See Also

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

### Examples

```
# Using the \code{\link[spatstat.data]{chorley}} dataset
data(chorley)
f1 <- jitter_power(obs_data = unique(chorley),
                  sim_total = 2,
                  samp_control = "MVN",
                  s_control = 0.01,
                  verbose = FALSE
                  )
```

---

spatial\_data

*Simulate random data for SRR function*

---

### Description

Generate random bivariate data for a spatial relative risk function.

### Usage

```
spatial_data(
  win = spatstat::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 ( <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 realisation 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 <code>win</code> . 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 <code>win</code> . 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 <code>win</code> . 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 <code>win</code> . 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> .

<code>l_control</code>	Optional. A single positive number, a vector of positive numbers, a function( $x, y, \dots$ ), 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> .
<code>e_control</code>	Optional. A single non-negative number for the size of the expansion of the simulation window for generating parent points. Ignored if <code>samp_control!="clustered"</code> .
<code>...</code>	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> .

### 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](#) 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 realisation 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,
  n_sim = 1,
  cols = c("#000000", "#cccccc", "#ffffff", "#ff0000", "#0000ff"),
  chars = c(1, 1),
  sizes = c(1, 1),
  plot_pts = TRUE,
  plot_text = FALSE,
  ...
)
```

**Arguments**

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

n_sim	Integer. The number of simulated iterations to plot. The default is one (1).
cols	Character string of length five (5) specifying the colors for plotting: 1) sufficiently powered, midpoint, insufficiently powered, case locations, control locations. The default colors in hex are <code>c("#0000ff", "#00ff00", "#ff0000", "#a020f0", "#ffa500")</code> or <code>c("grey0", "grey80", "grey100", "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> .
plot_pts	Logical. If TRUE (the default), the points from the first simulation iteration will be added to second plot. Not if FALSE.
plot_text	Logical. If TRUE, the local statistical power will be printed at each grid cell. Not if FALSE (the default).
...	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_data` function, this function will only display the first plot.

### Examples

```
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::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,
    lower_tail = 0.025,
    upper_tail = 0.975,
    cascon = FALSE,
    verbose = TRUE,
    parallel = FALSE,
    n_core = 2,
    ...
)

```

### 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 realisation 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 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".
lower_tail	Optional. Numeric value of lower p-value threshold (default=0.025).
upper_tail	Optional. Numeric value of upper p-value threshold (default=0.975). Ignored if cascon=FALSE.
cascon	Logical. If TRUE, computes the statistical power to detect case clusters and control clusters. If FALSE (the default), computes the statistical power to detect case clusters only.
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 <a href="#">runifdisc</a> , <a href="#">disc</a> , <a href="#">rpoispp</a> , <a href="#">rsyst</a> , or <a href="#">rNeymanScott</a> depending on samp_control or samp_control. Arguments also passed to <a href="#">risk</a> to select bandwidth, edge correction, and resolution.

## 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](#) 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  $\lambda = n\_control / [\text{resolution} \times \text{resolution}]$ . 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 realisation 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 "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  $[\text{resolution} \times \text{resolution}]$  of the mean relative risk values at each gridded knot.

`pval_mean` Vector of length  $[\text{resolution} \times \text{resolution}]$  of the mean asymptotic p-value at each gridded knot.

`rr_sd` Vector of length  $[\text{resolution} \times \text{resolution}]$  of the standard deviation of relative risk values at each gridded knot.

`rr_mean` Vector of length  $[\text{resolution} \times \text{resolution}]$  of the proportion of asymptotic p-values that were significant at each gridded knot.

`rx` Vector of length  $[\text{resolution} \times \text{resolution}]$  of the x-coordinates of each gridded knot.

`ry` Vector of length  $[\text{resolution} \times \text{resolution}]$  of the y-coordinates of each gridded knot.

`rx` 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.

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

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

### 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

`as.owin`, [5](#), [9](#)

`disc`, [6](#), [7](#), [10](#)

`image.plot`, [8](#)

`jitter_power`, [2](#), [7](#)

`plot.ppp`, [8](#)

`risk`, [3](#), [4](#), [6](#), [10](#), [11](#)

`rNeymanScott`, [6](#), [7](#), [10](#)

`rpoispp`, [6](#), [7](#), [10](#)

`rsyst`, [6](#), [7](#), [10](#)

`runifdisc`, [6](#), [7](#), [10](#)

`spatial_data`, [4](#), [7](#), [8](#)

`spatial_plots`, [7](#)

`spatial_power`, [7](#), [8](#)

`spatstat`, [6](#), [10](#)