

# Package ‘onpoint’

February 2, 2022

**Type** Package

**Title** Helper Functions for Point Pattern Analysis

**Version** 1.0.2

**Maintainer** Maximillian H.K. Hesselbarth <mhk.hesselbarth@gmail.com>

**Description** Growing collection of helper functions for point pattern analysis. Most functions are designed to work with the 'spatstat' (<<http://spatstat.org>>) package. The focus of most functions are either null models or summary functions for spatial point patterns. For a detailed description of all null models and summary functions, see Wiegand and Moloney (2014, ISBN:9781420082548).

**URL** <https://r-spatialecology.github.io/onpoint/>

**BugReports** <https://github.com/r-spatialecology/onpoint/issues>

**License** GPL (>= 3)

**Depends** R (>= 3.1)

**Imports** ggplot2, spatstat.core, spatstat.geom, spatstat.random, stats

**Suggests** covr, spatstat (>= 2.0.0), testthat

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Maximillian H.K. Hesselbarth [aut, cre]  
(<<https://orcid.org/0000-0003-1125-9918>>)

**Repository** CRAN

**Date/Publication** 2022-02-02 21:10:02 UTC

## R topics documented:

balance_points . . . . .	2
center_l_function . . . . .	3
estimate_o_ring . . . . .	4
estimate_pcf_fast . . . . .	5
plot.env_summarized . . . . .	6

plot_quantums . . . . .	7
print.env_summarized . . . . .	9
rlabel_local . . . . .	10
simulate_antecedent_conditions . . . . .	11
simulate_heterogenous_pattern . . . . .	12
summarize_envelope . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

balance_points	<i>balance_points</i>
----------------	-----------------------

---

## Description

Balance number of points

## Usage

```
balance_points(pattern, n, verbose = TRUE)
```

## Arguments

pattern	ppp object.
n	Either an integer or a ppp object.
verbose	Print messages.

## Details

The function balances out the number of points in the input pattern to either the provided number of points as integer or the same number of points if a ppp object is provided.

## Value

ppp

## Examples

```
set.seed(42)
input <- spatstat.random::rpoispp(lambda = 100)
input_b <- spatstat.random::rpoispp(lambda = 100)

balance_points(pattern = input, n = 110)
balance_points(pattern = input, n = input_b)
```

---

center\_l\_function      *center\_l\_function*

---

**Description**

Centered L-function

**Usage**

```
center_l_function(x, ...)
```

**Arguments**

x	ppp
...	Arguments passed to <code>spatstat.core::Lest()</code>

**Details**

Centers Besag's L-function to zero by calculating  $L(r) - r$ . Centering the L-function allows an easier interpretation and plotting of the results (Haase 1995).

Returns an 'Function value object' of the spatstat package.

**Value**

fv.object

**References**

Besag, J.E., 1977. Discussion on Dr. Ripley's paper. *Journal of the Royal Statistical Society. Series B (Methodological)* 39, 193–195. <<https://doi.org/10.1111/j.2517-6161.1977.tb01616.x>>

Ripley, B.D., 1977. Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B (Methodological)* 39, 172–192. <<https://doi.org/10.1111/j.2517-6161.1977.tb01615.x>>

Haase, P., 1995. Spatial pattern analysis in ecology based on Ripley's K-function: Introduction and methods of edge correction. *Journal of Vegetation Science* 6, 575–582. <<https://doi.org/10.2307/3236356>>

**See Also**

[Lest](#)

**Examples**

```
input_pattern <- spatstat.random::runifpoint(n = 100)
center_l_function(input_pattern, correction = "Ripley")
```

```
lest <- spatstat.core::Lest(input_pattern)
center_l_function(lest)
```

---

estimate_o_ring	<i>estimate_o_ring</i>
-----------------	------------------------

---

**Description**

O-ring function

**Usage**

```
estimate_o_ring(x, ...)
```

**Arguments**

x	ppp
...	Arguments passed to <code>spatstat.core::pcf.ppp()</code>

**Details**

Estimates the O-ring function proposed by Wiegand and Moloney (2004). The O-ring statistic is defined as:

$$O(r) = \lambda * g(r)$$

Generally speaking,  $O(r)$  scales the pair correlation  $g(r)$  function with help of the intensity  $\lambda$ . One advantage of the O-ring statistic is that it can be interpreted as a neighborhood density because it is a probability density function (Wiegand & Moloney 2004, 2014).

Returns an 'Function value object' of the spatstat package.

**Value**

fv.object

**References**

Wiegand, T., Moloney, K.A., 2004. Rings, circles, and null models for point pattern analysis in ecology. *Oikos* 104, 209–229. <<https://doi.org/10.1111/j.0030-1299.2004.12497.x>>

Wiegand, T., Moloney, K.A., 2014. Handbook of spatial point-pattern analysis in ecology. Chapman and Hall/CRC Press, Boca Raton, USA. <isbn:978-1-4200-8254-8>

**See Also**

[density.ppp](#)  
[pcf](#)

**Examples**

```
input_pattern <- spatstat.random::runifpoint(n = 100)
estimate_o_ring(input_pattern)
```

---

estimate_pcf_fast	<i>estimate_pcf_fast</i>
-------------------	--------------------------

---

## Description

Fast estimation of the pair correlation function

## Usage

```
estimate_pcf_fast(pattern, ...)
```

## Arguments

pattern	Point pattern.
...	Arguments passed down to 'Kest' or 'pcf.fv'.

## Details

The functions estimates the pair correlation functions based on an estimation of Ripley's K-function. This makes it computationally faster than estimating the pair correlation function directly.

It is a wrapper around Kest and pcf.fv and returns a 'Function value object' of the spatstat package.

## Value

fv.object

## References

Ripley, B.D., 1977. Modelling spatial patterns. Journal of the Royal Statistical Society. Series B (Methodological) 39, 172–192. <<https://doi.org/10.1111/j.2517-6161.1977.tb01615.x>>

Stoyan, D., Stoyan, H., 1994. Fractals, random shapes and point fields. John Wiley & Sons, Chichester, UK. <isbn:978-0-471-93757-9>

## See Also

[Kest](#)  
[pcf.fv](#)

## Examples

```
set.seed(42)
pattern <- spatstat.random::runifpoint(n = 100)
pcf_fast <- estimate_pcf_fast(pattern)
```

---

plot.env\_summarized    *plot.env\_summarized*

---

### Description

Plotting method for env\_summarized object

### Usage

```
## S3 method for class 'env_summarized'  
plot(  
  x,  
  col = c("#97CBDE", "#E1B0B5"),  
  x_lab = NULL,  
  y_lab = NULL,  
  base_size = 10,  
  label = TRUE,  
  ...  
)
```

### Arguments

x	Random patterns.
col	Colors for areas above and below envelope.
x_lab, y_lab	Labels of x- and y-axis.
base_size	Base size of plot
label	If TRUE the ratios of the area above and below are added to the plot.
...	To be generic for plotting function.

### Details

Plotting method for summarized envelope created with [summarize\\_envelope](#).

Returns a ggplot object.

### Value

ggplot

### See Also

[summarize\\_envelope](#)

## Examples

```
set.seed(42)
input_pattern <- spatstat.random::rThomas(kappa = 15, scale = 0.05, mu = 5)

cluster_env <- spatstat.core::envelope(input_pattern, fun = "pcf", nsim = 39,
funargs = list(divisor = "d", correction = "Ripley", stoyan = 0.25))

x <- summarize_envelope(cluster_env)
plot(x)
```

---

plot\_quantums                      *plot\_quantums*

---

## Description

Plot simulation envelopes

## Usage

```
plot_quantums(
  input,
  labels = NULL,
  color_scale = NULL,
  legend_position = "bottom",
  quantum_position = NULL,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  line_size = 0.5,
  base_size = 15,
  full_fun = TRUE,
  quantum = TRUE,
  standarized = FALSE
)
```

## Arguments

input	envelope.
labels	Name of the labels. See details for more information.
color_scale	Colors used with labels.
legend_position	The position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)
quantum_position	Vector with minimum and maximum y value of the quantum bar.

title	Plot title.
xlab, ylab	axis labels.
line_size	Size of the lines.
base_size	Base font size.
full_fun	If true observed value and envelope is plotted.
quantum	If true quantum bars are plotted.
standarized	If true observed value is standardized. See details for more details.

### Details

This functions provides a plotting style for envelope objects of the spatstat package (for more information please see `spatstat.core::envelope`). The location of the observed value in relation to the simulation envelope of the null model input is indicated by an additional colour bar at the bottom of the plot. If `standarized = TRUE`, all values are standarized by subtracting the theoretical value for CSR

Labels must be a vector including labels for the following three cases. The color scale vector is used in the same order.

1 = observed > high

2 = low < observed < high

3 = observed < low

To adjust the position of the quantum bar, use `quantum_position`.

Returns a ggplot object.

### Value

ggplot

### References

Esser, D.S., Leveau, J.H.J., Meyer, K.M., Wiegand, K., 2015. Spatial scales of interactions among bacteria and between bacteria and the leaf surface. *FEMS Microbiology Ecology* 91, 1–13. <<https://doi.org/10.1093/femsec/fi>>

### See Also

[envelope](#)

### Examples

```
set.seed(42)
pattern <- spatstat.random::rThomas(kappa = 50, scale = 0.025, mu = 5)
csr_envelope <- spatstat.core::envelope(pattern, fun = spatstat.core::pcf, nsim = 19)
plot_quantums(csr_envelope, ylab = "g(r)")
```



---

```
print.env_summarized  print.env_summarized
```

---

## Description

Print method for env\_summarized object

## Usage

```
## S3 method for class 'env_summarized'  
print(x, return_area = FALSE, digits = 2, ...)
```

## Arguments

x	Random patterns.
return_area	If true, not the ratio but the area is returned.
digits	Number of decimal places (round).
...	Arguments passed to cat

## Details

Printing method for summarized envelope created with [summarize\\_envelope](#).

## Value

No return value

## See Also

[summarize\\_envelope](#)

## Examples

```
set.seed(42)  
input_pattern <- spatstat.random::rThomas(kappa = 15, scale = 0.05, mu = 5)  
  
cluster_env <- spatstat.core::envelope(input_pattern, fun = "pcf", nsim = 39,  
  funargs = list(divisor = "d", correction = "Ripley", stoyan = 0.25))  
  
x <- summarize_envelope(cluster_env)  
print(x)
```

---

<code>rlabel_local</code>	<i>rlabel_local</i>
---------------------------	---------------------

---

**Description**

Local random labelling of marked point pattern

**Usage**

```
rlabel_local(X, distance, nsim = 19, drop = TRUE, verbose = TRUE)
```

**Arguments**

<code>X</code>	ppp
<code>distance</code>	Mark of points that do not change.
<code>nsim</code>	Number of patterns to simulate.
<code>drop</code>	If <code>nsim = 1</code> and <code>drop = TRUE</code> , the result will be a point pattern, rather than a list containing a point pattern.
<code>verbose</code>	If <code>TRUE</code> warning messages are printed.

**Details**

Local random labelling function, i.e. marks will be shuffled only across points within the specified local distance. Technically, this is achieved by sampling the mark of a neighbouring point  $j$  within the distance  $d$  for the focal point  $i$ . Thus, the distance  $d$  must be selected in a way that each point has at least one neighbour within  $d$ .

Returns a list with ppp objects.

**Value**

list

**References**

Velázquez, E., Martínez, I., Getzin, S., Moloney, K.A., Wiegand, T., 2016. An evaluation of the state of spatial point pattern analysis in ecology. *Ecography* 39, 1–14. <<https://doi.org/10.1111/ecog.01579>>

Wiegand, T., Moloney, K.A., 2014. Handbook of spatial point-pattern analysis in ecology. Chapman and Hall/CRC Press, Boca Raton, USA. <isbn:978-1-4200-8254-8>

**See Also**

[rlabel](#)

**Examples**

```

set.seed(42)
pattern <- spatstat.random::runifpoint(n = 250, win = spatstat.geom::owin(c(0, 100), c(0, 100)))
spatstat.geom::marks(pattern) <- runif(n = 250, min = 10, max = 120)

rlabel_local(X = pattern, distance = 25, nsim = 19)

```

---

```

simulate_antecedent_conditions
      simulate_antecedent_conditions

```

---

**Description**

Simulate heterogenous pattern

**Usage**

```
simulate_antecedent_conditions(x, i, j, nsim, heterogenous = FALSE, ...)
```

**Arguments**

x	ppp
i	Mark of points that are randomized.
j	Mark of points that do not change.
nsim	Number of patterns to simulate.
heterogenous	If TRUE, points with the mark i are randomized using a heterogeneous Poisson process.
...	Arguments passed to spatstat.core::density.ppp().

**Details**

Simulate point patterns as null model data for `spatstat.core::envelope()` using antecedent conditions as null model. `x` must be marked point pattern. Antecedent conditions are suitable as a null model if points of type `j` may influence points of type `i`, but not the other way around (Velazquez et al 2016). One example are the positions of seedlings that may be influenced by the position of mature trees.

Returns a list with ppp objects.

**Value**

list

## References

- Velázquez, E., Martínez, I., Getzin, S., Moloney, K.A., Wiegand, T., 2016. An evaluation of the state of spatial point pattern analysis in ecology. *Ecography* 39, 1–14. <<https://doi.org/10.1111/ecog.01579>>
- Wiegand, T., Moloney, K.A., 2014. Handbook of spatial point-pattern analysis in ecology. Chapman and Hall/CRC Press, Boca Raton, USA. <isbn:978-1-4200-8254-8>

## See Also

[envelope](#)

## Examples

```
set.seed(42)
pattern_a <- spatstat.random::runifpoint(n = 20)
spatstat.geom::marks(pattern_a) <- "a"
pattern_b <- spatstat.random::runifpoint(n = 100)
spatstat.geom::marks(pattern_b) <- "b"
pattern <- spatstat.geom::superimpose(pattern_a, pattern_b)

null_model <- simulate_antecedent_conditions(x = pattern, i = "b", j = "a", nsim = 19)
spatstat.core::envelope(Y = pattern, fun = spatstat.core::pcf, nsim = 19, simulate = null_model)
```

---

```
simulate_heterogenous_pattern
      simulate_heterogenous_pattern
```

---

## Description

Simulate heterogeneous pattern

## Usage

```
simulate_heterogenous_pattern(x, nsim, fix_n = FALSE, ...)
```

## Arguments

<code>x</code>	ppp
<code>nsim</code>	Number of patterns to simulate.
<code>fix_n</code>	Logical if true the null model patterns have exactly the same number of points as input.
<code>...</code>	Arguments passed to <code>spatstat.core::density.ppp()</code> .

## Details

Simulate heterogeneous point patterns as null model data for `spatstat.core::envelope()`. A heterogeneous Poisson process is used, meaning that there are no interaction between points, however, the simulated coordinates depend on the intensity  $\lambda$  of the input pattern.

Returns a list with ppp objects.

## Value

list

## References

Baddeley, A., Rubak, E., Turner, R., 2015. Spatial point patterns: Methodology and applications with R. Chapman and Hall/CRC Press, London, UK. <isbn:978-1-4822-1020-0>

Wiegand, T., Moloney, K.A., 2014. Handbook of spatial point-pattern analysis in ecology. Chapman and Hall/CRC Press, Boca Raton, USA. <isbn:978-1-4200-8254-8>

## See Also

[envelope](#)  
[density.ppp](#)

## Examples

```
set.seed(42)
input_pattern <- spatstat.random::rpoispp(lambda = function(x , y) {100 * exp(-3 * x)}), nsim = 1)
null_model <- simulate_heterogenous_pattern(input_pattern, nsim = 19)
spatstat.core::envelope(Y = input_pattern, fun = spatstat.core::pcf, nsim = 19,
simulate = null_model)
```

---

`summarize_envelope`      *summarize\_envelope*

---

## Description

Summarize simulation envelope

## Usage

```
summarize_envelope(x, plot_result = FALSE)
```

## Arguments

<code>x</code>	fv
<code>plot_result</code>	A plot is drawn.

**Details**

The area above and below the null model envelope is divided by the total area under the curve. If `seperated = TRUE`, the first returning value is the relative area above, the second value the relative value below the envelope. If `seperated = FALSE` the value is the absolute sum of both ratio. If the value is positive, the area above the envelope is larger than the value below the envelope. If the value is negative, the area under the envelope is larger than the value above the envelope.

The returned `env_summarized` object includes information about the area under the curve where the summary function observed pattern is above or below the null model envelopes.

**Value**

`env_summarized`

**See Also**

[envelope](#)

**Examples**

```
set.seed(42)
input_pattern <- spatstat.random::rThomas(kappa = 15, scale = 0.05, mu = 5)

cluster_env <- spatstat.core::envelope(input_pattern, fun = "pcf", nsim = 39,
funargs = list(divisor = "d", correction = "Ripley", stoyan = 0.25))

summarize_envelope(cluster_env)
```

# Index

balance\_points, [2](#)

center\_l\_function, [3](#)

density.ppp, [4](#), [13](#)

envelope, [8](#), [12–14](#)

estimate\_o\_ring, [4](#)

estimate\_pcf\_fast, [5](#)

Kest, [5](#)

Lest, [3](#)

pcf, [4](#)

pcf.fv, [5](#)

plot.env\_summarized, [6](#)

plot\_quantums, [7](#)

print.env\_summarized, [9](#)

rlabel, [10](#)

rlabel\_local, [10](#)

simulate\_antecedent\_conditions, [11](#)

simulate\_heterogenous\_pattern, [12](#)

summarize\_envelope, [6](#), [9](#), [13](#)