Package: apcf (via r-universe)

October 23, 2026

Title Adapted Pair Correlation Function

Version 0.3.3

Description The adapted pair correlation function transfers the concept of the pair correlation function from point patterns to patterns of objects of finite size and irregular shape (e.g. lakes within a country). The pair correlation function describes the spatial distribution of objects, e.g. random, aggregated or regularly spaced. This is a reimplementation of the method suggested by Nuske et al. (2009) <doi:10.1016/j.foreco.2009.09.050> using the library 'GEOS' <doi:10.5281/zenodo.11396894>.

License GPL (>= 3)

URL https://rnuske.github.io/apcf/, https://github.com/rnuske/apcf

BugReports https://github.com/rnuske/apcf/issues

Encoding UTF-8

Depends R (>= 3.3.0)

Imports Rcpp (>= 0.12), wk (>= 0.6.0)

Suggests knitr, rmarkdown

LinkingTo Rcpp

SystemRequirements GEOS (>= 3.4.0)

VignetteBuilder knitr

LazyData true

RoxygenNote 7.3.3

NeedsCompilation yes

Author Robert Nuske [aut, cre] (ORCID: <https://orcid.org/0000-0001-9773-2061>)

Maintainer Robert Nuske <robert.nuske@mailbox.org>

Config/pak/sysreqs libgeos-dev

Repository https://cran.r-universe.dev

Date/Publication 2025-12-21 13:10:02 UTC

RemoteUrl https://github.com/cran/apcf

RemoteRef HEAD

RemoteSha 105d5a4ae3accfb6976658d4d1bfbf87138a6937

Contents

apcf-package 2
dists 3
dists2pcf 4
fv_pcf 6
pat2dists 7
plot.fv_pcf 8
sim_patterns 10

Index **12**

`apcf-package` *Adapted Pair Correlation Function*

Description

A faster implementation of the Adapted Pair Correlation Function presented in Nuske et al. (2009) in C++ using the library GEOS directly instead of through PostGIS.

Details

The Adapted Pair Correlation Function transfers the concept of the Pair Correlation Function from point patterns to patterns of objects of finite size and irregular shape (eg. lakes within a country). The main tasks are (i) the construction of null models by rondomizing the objects of the original pattern within the study area, (ii) the edge correction by determining the proportion of a buffer within the study area, and (iii) the calculation of the shortest distances between the objects.

This package mainly provides three functions:

- `pat2dists()` creates null models and calculates distances and ratios,
- `dists2pcf()` turns distances and ratios into an edge corrected PCF, and
- `plot()` plots Pair Correlation Functions.

Pattern to Distances & Ratios

The task consists of two parts: creating null models / permutations and calculating distances between all objects of a pattern and determining the fraction of the perimeter a buffer inside the study area. Permutations of the original pattern are achieved by randomly rotating and randomly placing all objects within the study area without overlap.

The resulting collection of distances and ratios of each null model and the original pattern are returned as an object of class dists (a data.frame with some additional attributes).

The library GEOS (>= 3.4.0) is used for the geometrical analysis of the pattern. Geodata are converted to GEOS Geometries. The GEOS functions are called from C++ Functions which are integrated into R via Rcpp and wrapped in the R function `pat2dists()`.

Create an edge corrected PCF

The dists objects are turned into fv_pcf objects by the function `dists2pcf()`. A C++ function finds all distances and ratios belonging to a null model or the original pattern (marked with index 0) and calculates a density function using the Epanechnikov kernel and Ripley's edge correction. Resulting in as many PCFs as null models were created plus a PCF for the original pattern. From the PCF of the null models a pointwise critical envelope is derived. The arithmetic mean of all PCF of the null models is employed for a bias correction of the empirical PCF and the upper and lower bound of the envelope.

Plot a PCF

`plot.fv_pcf()` is an S3 method of the plot function for the class fv_pcf. It provides a nice plot of the empirical PCF together with the pointwise critical envelope.

Author(s)

Maintainer: Robert Nuske <robert.nuske@mailbox.org> (ORCID)

References

Nuske, R.S., Sprauer, S. and Saborowski, J. (2009): Adapting the pair-correlation function for analysing the spatial distribution of canopy gaps. Forest Ecology and Management, 259(1): 107–116. https://doi.org/10.1016/j.foreco.2009.09.050

See Also

Useful links:

- https://rnuske.github.io/apcf/
- https://github.com/rnuske/apcf
- Report bugs at https://github.com/rnuske/apcf/issues

---

dists *Class dists: Collection of Distances and Ratios*

Description

Advanced Use Only. This low-level function creates an object of class "dists" from raw numerical data.

Usage

```
dists(df, area, n_obj, max_dist)

is.dists(obj)
```

Arguments

df	A data frame with the columns sim, dist, and ratio containing an indicator of the model run (0:n_sim), distances between the objects of the patterns, and the ratios of a buffer with distance dist inside the study area (needed for Ripley's edge correction).
area	size of the study area in square units
n_obj	number of objects in the pattern
max_dist	maximum distance to be measured in pattern
obj	an R object, preferably of class dists

Value

An object of class dists.

---

dists2pcf *Convert Distances & Ratios to PCF*

Description

Estimates the Adapted Pair Correlation Function (PCF) of a pattern together with a pointwise critical envelope based on distances and ratios calculated by pat2dists().

Usage

```
dists2pcf(dists, r, r_max = NULL, kernel = "epanechnikov", stoyan, n_rank)
```

Arguments

dists	An object of class dists. Usually created by pat2dists()
r	A step size or a vector of values for the argument r at which g(r) should be evaluated.
r_max	maximum value for the argument r.
kernel	String. Choice of smoothing kernel (only the "epanechnikov" kernel is currently implemented).
stoyan	Bandwidth coefficient (smoothing the Epanechnikov kernel). Penttinen et al. (1992) and Stoyan and Stoyan (1994) suggest values between 0.1 and 0.2.
n_rank	Rank of the value amongst the n_sim simulated values used to construct the envelope. A rank of 1 means that the minimum and maximum simulated values will be used. Must be >= 1 and < n_sim/2. Determines together with n_sim in pat2dists() the alpha level of the envelope. If alpha and n_sim are fix, n_rank can be calculated by (n_sim+1)*alpha/2 eg. (199+1) * 0.05/2 = 5

Details

Since the pair-correlation function is a density function, we employ the frequently used Epanechnikov kernel (Silverman 1986, Stoyan and Stoyan 1994, Nuske et al. 2009). The Epanechnikov kernel is a weight function putting maximal weight to pairs with distance exactly equal to *r* but also incorporating pairs only roughly at distance *r* with reduced weight. This weight falls to zero if the actual distance between the points differs from *r* by at least δ, the so-called bandwidth parameter, which determines the degree of smoothness of the function. Penttinen et al. (1992) and Stoyan and Stoyan (1994) suggest to set *c* aka stoyan-parameter of $c/\sqrt{\lambda}$ between 0.1 and 0.2 with λ being the intensity of the pattern.

The edge correction is based on suggestions by Ripley (1981). For each pair of objects i and j, a buffer with buffer distance r_{ij} is constructed around the object i. The object j is then weighted by the inverse of the ratios p_{ij} of the buffer perimeter being within the study area. That way we account for the reduced probability of finding objects close to the edge of the study area.

The alpha level of the pointwise critical envelope is $\alpha = \frac{n_rank*2}{n_sim+1}$ according to (Besag and Diggle 1977, Buckland 1984, Stoyan and Stoyan 1994).

Value

An object of class fv_pcf containing the function values of the PCF and the envelope.

References

Besag, J. and Diggle, P.J. (1977): Simple Monte Carlo tests for spatial pattern. Journal of the Royal Statistical Society. Series C (Applied Statistics), 26(3): 327–333. https://doi.org/10.2307/2346974

Buckland, S.T. (1984). Monte Carlo Confidence Intervals. Biometrics, 40(3): 811-817. https://doi.org/10.2307/2530926

Nuske, R.S., Sprauer, S. and Saborowski, J. (2009): Adapting the pair-correlation function for analysing the spatial distribution of canopy gaps. Forest Ecology and Management, 259(1): 107–116. https://doi.org/10.1016/j.foreco.2009.09.050

Penttinen A., Stoyan D., Henttonen H. M. (1992): Marked point processes in forest statistics. Forest Science, 38(4): 806–824. https://doi.org/10.1093/forestscience/38.4.806

Ripley, B.D. (1981): Spatial Statistics. John Wiley & Sons, New York. https://doi.org/10.1002/0471725218

Silverman, B.W. (1986): Density Estimation for Statistics and Data Analysis. Chapman and Hall, London.

Stoyan, D. and Stoyan, H. (1994) Fractals, random shapes and point fields: Methods of geometrical statistics. John Wiley & Sons, Chichester.

See Also

`pat2dists()`, `plot.fv_pcf()`

Examples

```
# it's advised against setting n_sim < 199
ds <- pat2dists(area=sim_area, pattern=sim_pat_reg, max_dist=25, n_sim=3)

# derive PCF and envelope
pcf <- dists2pcf(ds, r=0.2, r_max=25, stoyan=0.15, n_rank=1)
```

fv_pcf *Class fv_pcf: Function Value Table for PCFs*

Description

Advanced Use Only. This low-level function creates an object of class "fv_pcf" from raw numerical data.

Usage

```
fv_pcf(df, n_sim, n_rank, correc, kernel, stoyan, bw)

is.fv_pcf(obj)

## S3 method for class 'fv_pcf'
print(x, ...)

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

Arguments

df	A data frame with at least 2 columns named 'r' and 'g' containing the values of the function argument (r) and the corresponding values (g). Usually the upper 'upr' and lower 'lwr' bounds of a pointwise critical envelope are passed along as well.
n_sim	Integer. Number of generated simulated patterns used for computing the envelope
n_rank	Integer. Rank of the envelope value amongst the n_sim simulated values. A rank of 1 means that the minimum and maximum simulated values will be used.
correc	String. Choice of edge correction (eg. "Ripley").
kernel	String. Choice of smoothing kernel (eg. "epanechnikov").
stoyan	Bandwidth coefficient used in smoothing kernel.
bw	Bandwidth used in smoothing kernel.
x, obj, object	an R object, preferably of class fv_pcf
...	additional parameter

Value

An object of class fv_pcf.

pat2dists *Convert a Pattern to Distances & Ratios*

Description

Creates n_sim null models by permutation of the original pattern and calculates distances between all object of a pattern closer than max_dist and determines the fractions of the perimeter of buffers with distance dist inside the study area (needed for edge correction).

Usage

```
pat2dists(
  area,
  pattern,
  max_dist,
  n_sim = 199,
  max_tries = 1e+05,
  save_pattern = FALSE,
  verbose = FALSE
)
```

Arguments

area, pattern	Geodata (polygons) of study area and pattern in the formats WKB (well known binary, list of raw vectors), WKT (well known text) or sf-objects if package sf is available. Via sf all file formats supported by GDAL/OGR are possible.
max_dist	Maximum distance measured in the pattern. Usually smaller than half the diameter of the study area.
n_sim	Number of simulated patterns (randomizations) to be generated for computing the envelope and correcting the biased empirical pcf. Determines together with n_rank in dists2pcf() the alpha level of the envelope. If alpha and n_rank are fix, n_sim can be calculated by (n_rank*2/alpha)-1 for instance (5*2/0.05)-1 = 199.
max_tries	How often shall a relocation of an object be tried while randomizing the pattern.
save_pattern	Shall one simulated pattern be returned in the attributes for debugging/later inspections. The pattern is provided as WKB (list of raw vectors) in the attribute randPattern.
verbose	Provide progress information in the console. Roman numerals (x: 10, C: 100, D: 500, M: 1000) indicate the progress of the simulation and 'e' denotes the empirical PCF.

Details

Null models are created by randomly rotating and randomly placing all objects within the study area without overlap. They are used for correcting the biased pcf and constructing a pointwise critical envelope (cf. Nuske et al. 2009).

Measuring distances between objects and permutation of the pattern is done using GEOS.

Value

An object of class dists. If `save_pattern = TRUE` an additional attribute `randPattern` is returned containing a WKB (list of raw vectors).

References

Nuske, R.S., Sprauer, S. and Saborowski, J. (2009): Adapting the pair-correlation function for analysing the spatial distribution of canopy gaps. Forest Ecology and Management, 259(1): 107–116. https://doi.org/10.1016/j.foreco.2009.09.050

See Also

`dists2pcf()`, `plot.fv_pcf()`

Examples

```
# it's advised against setting n_sim < 199
ds <- pat2dists(area=sim_area, pattern=sim_pat_reg, max_dist=25, n_sim=3)

# verbose and returns one randomized pattern for debugging
ds_plus <- pat2dists(area=sim_area, pattern=sim_pat_reg, max_dist=5, n_sim=3,
                     verbose=TRUE, save_pattern=TRUE)

## Not run:
  # wk's plot function needs additional package 'vctrs'
  plot(attr(ds_plus, "randPattern"))

## End(Not run)
```

`plot.fv_pcf` *Plot PCF*

Description

Plot method for the class "fv_pcf". Draws a pair correlation function and a pointwise critical envelope if available.

Usage

```
## S3 method for class 'fv_pcf'
plot(
  x,
  xlim = NULL,
  ylim = NULL,
  xticks = NULL,
  yticks = NULL,
  xlab = "r",
  ylab = "g(r)",
```

```
  main = NULL,
  sub = NULL,
  xaxis = TRUE,
  yaxis = TRUE,
  ann = graphics::par("ann"),
  bty = "l",
  ...
)
```

Arguments

x	an object of class fv_pcf.
xlim, ylim	the x and y limits of the plot. NULL indicates that the range of the finite values to be plotted should be used.
xticks, yticks	points at which tick-marks are to be drawn. By default (when NULL) tickmark locations are computed
xlab, ylab	a label for the x and y axis, respectively.
main, sub	a main and sub title for the plot, see also `title()`.
xaxis, yaxis	a logical value or a 1-character string giving the desired type of axis. The following values are possible: "n" or FALSE for no axis, "t" for ticks only, "f" or TRUE for full axis, "o" for full axis in the outer margin.
ann	a logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot.
bty	a character string which determines the type of box which is drawn about the plotting region, see `par()`.
...	additional parameter, currently without effect

Value

An object of class fv_pcf invisibly.

See Also

`pat2dists()`, `dists2pcf()`

Examples

```
# it's advised against setting n_sim < 199
ds <- pat2dists(area=sim_area, pattern=sim_pat_reg, max_dist=25, n_sim=3)

# derive PCF and envelope
pcf <- dists2pcf(ds, r=0.2, r_max=25, stoyan=0.15, n_rank=1)

# a simple plot
plot(x=pcf, xlim=c(0, 20), ylim=c(0, 2.2))

# a panel of four plots
op <- par(mfrow=c(2,2), oma=c(3,3,0,0), mar=c(0,0,2,2),
```

```
          mgp=c(2,0.5,0), tcl=-0.3)
plot(pcf, xaxis='t', yaxis='o', ann=FALSE)
plot(pcf, xaxis='t', yaxis='t', ann=FALSE)
plot(pcf, xaxis='o', yaxis='o', ann=FALSE)
plot(pcf, xaxis='o', yaxis='t')
par(op)
```

`sim_patterns` *Simulated Patterns (sample data)*

Description

The simulated patterns were created for testing the Adapted Pair Correlation Function presented in Nuske et al. (2009).

Usage

```
sim_area

sim_pat_clust

sim_pat_rand

sim_pat_reg
```

Format

A set of WKBs of class `wk_wkb` containing the study area and three simulated patterns.

Dataset name	Description
`sim_area`	study area
`sim_pat_reg`	simulated regular pattern
`sim_pat_rand`	simulated random pattern
`sim_pat_clust`	simulated clustered pattern

The study area is a square of 100 m x 100 m. A set of n = 100 objects were created and latter placed according to the designated spatial distribution. The size distribution and shapes of the objects are inspired by measurements of canopy gaps. The areas of the objects range from 1.6 m2 to 57.7 m2 with an arithmetic mean of 9.7 m2 and a median of 5.5 m2. The total area of all objects is 969.7 m2, meaning 9.7% of the study area is covered by objects.

For the `sim_pat_reg` dataset, the objects were arranged in a strict regular manner. A centric systematic grid was constructed, and the objects of the set were then randomly rotated and randomly placed by locating the centroids of the objects exactly on the matching randomly numbered grid points, resulting in a regular arrangement of objects with a constant distance of the centroids of 10 m.

For the sim_pat_rand dataset with randomly distributed objects, we generated a realisation of the Binomial process with intensity 0.01 m^-2, meaning one point per 100 m2. The objects were again randomly rotated and numbered and objects put on matching points with their centroid as close to the point as possible without overlapping other objects.

The sim_pat_clust dataset represents a clustered configuration. Again, we first created a point pattern with 100 points and then put the randomly numbered objects on the points. The point pattern was a realisation of Matern's cluster process with w = 0.0006 m^-2 or 6 cluster centres per ha, a dispersion radius of R = 10 m and on average y = 16.6 points per cluster.

We used the R-package spatstat (Baddeley et al. 2015) for simulating the Binomial process and Matern's cluster process.

Source

Nuske et al. 2009

References

Baddeley A., Rubak E. and Turner, R. (2015): Spatial Point Patterns: Methodology and Applications with R. Chapman and Hall/CRC, London. https://doi.org/10.1201/b19708

Nuske, R.S., Sprauer, S. and Saborowski, J. (2009): Adapting the pair-correlation function for analysing the spatial distribution of canopy gaps. Forest Ecology and Management, 259(1): 107–116. https://doi.org/10.1016/j.foreco.2009.09.050

Examples

```
ds <- pat2dists(area=sim_area, pattern=sim_pat_reg, max_dist=25, n_sim=3)
```

Index

∗ **datasets**
sim_patterns, 10

apcf *(*apcf-package*)*, 2
apcf-package, 2

dists, *2*, 3, *3, 4*, *8*
dists2pcf, 4
dists2pcf(), *2, 3*, *7–9*

fv_pcf, *3*, *5*, 6, *9*

is.dists *(*dists*)*, 3
is.fv_pcf *(*fv_pcf*)*, 6

par(), *9*
pat2dists, 7
pat2dists(), *2–5*, *9*
plot(), *2*
plot.fv_pcf, 8
plot.fv_pcf(), *3*, *5*, *8*
print.fv_pcf *(*fv_pcf*)*, 6

sim_area *(*sim_patterns*)*, 10
sim_pat_clust *(*sim_patterns*)*, 10
sim_pat_rand *(*sim_patterns*)*, 10
sim_pat_reg *(*sim_patterns*)*, 10
sim_patterns, 10
summary.fv_pcf *(*fv_pcf*)*, 6

title(), *9*

wk_wkb, *10*