

# Package: globalKinhom (via r-universe)

November 21, 2024

**Type** Package

**Title** Inhomogeneous K- And Pair Correlation Functions Using Global Estimators

**Version** 0.1.9

**Date** 2024-10-07

**Encoding** UTF-8

**Author** Thomas Shaw [aut, cre], Ege Rubak [ctb], Adrian Baddeley [ctb], Rolf Turner [ctb]

**Maintainer** Thomas Shaw <tshaw20@utk.edu>

**Depends** R (>= 3.5.0), spatstat.explore (>= 3.0)

**Imports** stats, utils, grDevices, spatstat.geom (>= 3.1), spatstat.random (>= 2.1.0), spatstat.univar

**Description** Second-order summary statistics K- and pair-correlation functions describe interactions in point pattern data. This package provides computations to estimate those statistics on inhomogeneous point processes, using the methods of in T Shaw, J Møller, R Waagepetersen, 2020 <doi:10.48550/arXiv.2004.00527>.

**License** GPL (>= 2)

**NeedsCompilation** yes

**Date/Publication** 2024-10-07 20:00:02 UTC

**Repository** <https://thomasrshaw.r-universe.dev>

**RemoteUrl** <https://github.com/cran/globalKinhom>

**RemoteRef** HEAD

**RemoteSha** d2f6b5a288a311d60e316e94078e941211b37122

## Contents

|                      |   |
|----------------------|---|
| globalKinhom-package | 2 |
| expectedPairs        | 3 |
| Kglobal              | 5 |
| pcfglobal            | 6 |

---

globalKinhom-package    *Inhomogeneous K- And Pair Correlation Functions Using Global Estimators*

---

## Description

Second-order summary statistics  $K$ - and pair-correlation functions describe interactions in point pattern data. This package provides computations to estimate those statistics on inhomogeneous point processes, using the methods of in T Shaw, J Møller, R Waagepetersen, 2020 <doi:10.48550/arXiv.2004.00527>.

## Details

The DESCRIPTION file:

```
Package:      globalKinhom
Type:         Package
Title:        Inhomogeneous K- And Pair Correlation Functions Using Global Estimators
Version:      0.1.9
Date:         2024-10-07
Encoding:     UTF-8
Authors@R:   c(person("Thomas", "Shaw", role=c("aut", "cre"), email="tshaw20@utk.edu"), person("Ege", "Rubak", role="
Author:       Thomas Shaw [aut, cre], Ege Rubak [ctb], Adrian Baddeley [ctb], Rolf Turner [ctb]
Maintainer:   Thomas Shaw <tshaw20@utk.edu>
Depends:      R (>= 3.5.0), spatstat.explore (>= 3.0)
Imports:      stats, utils, grDevices, spatstat.geom (>= 3.1), spatstat.random (>= 2.1.0), spatstat.univar
Description:  Second-order summary statistics K- and pair-correlation functions describe interactions in point pattern data. T
License:      GPL (>= 2)
```

Index of help topics:

```
Kglobal          (cross) K functions with a global intensity
                  reweighting
expectedPairs    Expected pairs in an inhomogeneous poisson
                  process
globalKinhom-package  Inhomogeneous K- And Pair Correlation Functions
                  Using Global Estimators
pcfglobal        (cross) pair correlation functions with a
                  global intensity reweighting
```

This package accompanies Shaw et al (2020). It provides “global” estimators for the non-parametric  $K$ - and pair correlation functions, which summarize the second order interactions of second-order intensity-reweighted stationary (SOIRS) point processes. These estimators provide an alternative to those proposed by Baddeley et al (2000) for SOIRS point processes, which we refer to as “local” estimators. The local estimators are implemented in the [spatstat.explore](#) package as [pcfinhom](#) and [Kinhom](#), with [pcfcross.inhom](#) and [Kcross.inhom](#) for the corresponding cross-pcf and cross- $K$ -function.

Where possible, the interfaces are made to match those used by the [spatstat.explore](#) package.



**Arguments**

|                 |                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rho1, rho2, rho | Intensity functions, either of class <code>im</code> or <code>funxy</code> . This may be produced by <a href="#">density.ppp</a> or <a href="#">densityfun.ppp</a> , or provided by a fitted intensity model.            |
| X               | Point pattern of class <code>ppp</code> with the points of the pattern for which $\bar{\gamma}$ is to be estimated.                                                                                                      |
| hx, hy          | For <code>expectedPairs</code> and <code>expectedCrossPairs</code> (i.e. $\gamma(h)$ ), the displacements $h \in \mathbb{R}^2$ to evaluate $\gamma$ at. These can be in any format supported by <code>xy.coords</code> . |
| r               | For the isotropic versions $\gamma^{\text{iso}}(r)$ , the separations $r$ at which $\gamma^{\text{iso}}$ is to be evaluated.                                                                                             |
| method          | Either <code>mc</code> (the default) or <code>lattice</code> . Compute integral using monte-carlo or on a lattice.                                                                                                       |
| tol             | A tolerance for how precise the integral should be. This is compared to a standard error for the <code>mc</code> estimate.                                                                                               |
| sigma           | Smoothing bandwidth for direct kernel-based estimators $\bar{\gamma}$ .                                                                                                                                                  |
| leaveoneout     | Use leave-out estimators. This should generally be true except for the purpose of evaluating the bias of the standard estimators. See Shaw et al 2020 for details.                                                       |
| maxeval         | Maximum number of evaluations of rho per iteration. Prevents memory-related crashes that can occur.                                                                                                                      |
| maxsamp         | Maximum number of monte carlo samples per iteration. If this is too large, you may do more work than required to achieve <code>tol</code> .                                                                              |
| dx              | if <code>method=="lattice"</code> , a lattice spacing for the computation. defaults to <code>.01</code> .                                                                                                                |

**Value**

The return value is a numeric vector with length equal to the number of displacements `h` passed

**Author(s)**

Thomas Shaw <[shawtr@umich.edu](mailto:shawtr@umich.edu)>

**References**

T Shaw, J Møller, R Waagepetersen. 2020. "Globally Intensity-Reweighted Estimators for  $K$ - and pair correlation functions". [arXiv:2004.00527](https://arxiv.org/abs/2004.00527) [stat.ME].

**See Also**

[pcfglobal](#), [Kglobal](#), which use these functions to compute the normalization functions  $\gamma$ .

---

Kglobal *(cross) K functions with a global intensity reweighting*

---

### Description

Compute  $K_{\text{global}}$

### Usage

```
Kglobal(X, lambda=NULL, ..., sigma=bw.CvL(X), r=NULL, rmax=NULL, breaks=NULL,
        normtol=.005, discrete.lambda=FALSE,
        interpolate=TRUE, interpolate.fac=10, isotropic=TRUE,
        leaveoneout=TRUE, exp_prs=NULL,
        interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)
```

```
Kcross.global(X, Y, lambdaX=NULL, lambdaY=NULL, ..., sigma=bw.CvL(X), r=NULL,
              rmax=NULL, breaks=NULL, normtol=.005,
              discrete.lambda=FALSE, interpolate=TRUE, isotropic=TRUE,
              interpolate.fac=10, leaveoneout=TRUE, exp_prs=NULL,
              interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)
```

### Arguments

|                          |                                                                                                                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| X, Y                     | point process of type <a href="#">ppp</a> , on which to evaluate the (cross) $K$ -function                                                                                                                                                                  |
| lambda, lambdaX, lambdaY | intensity function estimates corresponding to X and Y. If omitted, intensity functions will be computed using <a href="#">density.ppp</a> or <a href="#">densityfun.ppp</a> (see <code>discrete.lambda</code> below)                                        |
| ...                      | extra args passed to <a href="#">density.ppp</a> or <a href="#">densityfun.ppp</a> , if applicable.                                                                                                                                                         |
| sigma                    | Bandwidth value to use for kernel-based intensity estimation, intensity functions and <code>exp_prs</code> are not provided by the user.                                                                                                                    |
| r                        | Values of $r$ to evaluate $K(r)$ at. If omitted, a sensible default is chosen, using the same conventions as <a href="#">Kest</a> and <a href="#">Kinhom</a> .                                                                                              |
| rmax                     | Maximum $r$ to evaluate $K(r)$ at. <code>rmax</code> is used to generate values for <code>r</code> , if omitted. If missing, a sensible default is chosen.                                                                                                  |
| breaks                   | For internal use only.                                                                                                                                                                                                                                      |
| normtol                  | A tolerance to use for <code>expectedPairs</code> or <code>expectedCrossPairs</code> when computing monte-carlo estimates of the normalizing factor $\gamma$ . Expressed as a maximum fractional standard error.                                            |
| discrete.lambda          | If TRUE, and intensity function(s) are not supplied, estimate intensities by interpolating the values on a discrete lattice (using <a href="#">interp.im</a> and <a href="#">density.ppp</a> ), instead of exactly (using <a href="#">densityfun.ppp</a> ). |
| interpolate              | If TRUE, evaluate the <code>expectedCrossPairs</code> on a lattice and interpolate, rather than at the exact displacements observed in the pattern.                                                                                                         |

|                                |                                                                                                                                                                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>interpolate.fac</code>   | If interpolate, the lattice spacing will be $\text{sigma}/\text{interpolate.fac}$ .                                                                                                                                                                                                                                         |
| <code>isotropic</code>         | Set to TRUE to use the isotropic estimators $\gamma_{\text{iso}}$ .                                                                                                                                                                                                                                                         |
| <code>leaveoneout</code>       | Use the leave-one-out estimator for $\gamma$ . See Shaw et al, 2020 for details.                                                                                                                                                                                                                                            |
| <code>exp_prs</code>           | A function that returns values for $\gamma_{\text{iso}}(r)$ . If $\gamma$ is known explicitly, or the same calculation is being used for several point patterns, it can be much faster to compute it once and provide the function as <code>exp_prs</code> , since the computation of $\gamma$ is usually the slowest part. |
| <code>interpolate.maxdx</code> | Upper bound on allowable lattice spacing for interpolation.                                                                                                                                                                                                                                                                 |
| <code>dump</code>              | For debugging purposes, include computed values of $\gamma$ with the output, as at <code>trs</code> .                                                                                                                                                                                                                       |

**Value**

The return value is an object of class `fv`, just as for `Kest` and `Kinhom`. The object contains columns `r`, `theo`, and `global`, corresponding respectively to the argument  $r$ , the theoretical values of  $K(r)$  for a Poisson process, and  $K_{\text{global}}(r)$ .

**Author(s)**

Thomas Shaw <shawtr@umich.edu>

**References**

T Shaw, J Møller, R Waagepetersen. 2020. “Globally Intensity-Reweighted Estimators for  $K$ - and pair correlation functions”. arXiv:2004.00527 [stat.ME].

**See Also**

[expectedPairs](#)

**Examples**

```
rho <- funxy(function(x,y) 80*(1+x), owin())
X <- rpoispp(rho)
K <- Kglobal(X)
#plot(K)
```

---

pcfglobal

*(cross) pair correlation functions with a global intensity reweighting*

---

**Description**

Compute  $g_{\text{global}}$  or  $c_{\text{global}}$

**Usage**

```
pcfglobal(X, lambda=NULL, ..., sigma=bw.CvL(X), r=NULL, rmax=NULL,
  kernel="epanechnikov", bw=NULL, stoyan=0.15, normtol=.005, ratio=FALSE,
  discrete.lambda=FALSE, divisor=c("r", "d"),
  leaveoneout=TRUE, interpolate=TRUE, interpolate.fac=10, exp_prs=NULL,
  interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)
```

```
pcfcross.global(X,Y, lambdaX=NULL, lambdaY=NULL, ...,
  sigma=bw.CvL(X), r=NULL, rmax=NULL, kernel="epanechnikov", bw=NULL,
  stoyan=0.15, normtol=.005, ratio=FALSE, discrete.lambda=FALSE,
  divisor=c("r", "d"), analytical=NULL, interpolate=TRUE,
  interpolate.fac=10, exp_prs=NULL,
  interpolate.maxdx=diameter(as.owin(X))/100, dump=FALSE)
```

**Arguments**

|                          |                                                                                                                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| X, Y                     | point process of type <a href="#">ppp</a> , on which to evaluate the (cross) $K$ -function                                                                                                                                                                  |
| lambda, lambdaX, lambdaY | intensity function estimates corresponding to X and Y. If omitted, intensity functions will be computed using <a href="#">density.ppp</a> or <a href="#">densityfun.ppp</a> (see <code>discrete.lambda</code> below)                                        |
| ...                      | extra args passed to <a href="#">density.ppp</a> or <a href="#">densityfun.ppp</a> , if applicable.                                                                                                                                                         |
| sigma                    | Bandwidth value to use for kernel-based intensity estimation, intensity functions and <code>exp_prs</code> are not provided by the user.                                                                                                                    |
| r                        | Values of $r$ to evaluate $K(r)$ at. If omitted, a sensible default is chosen, using the same conventions as <a href="#">Kest</a> and <a href="#">Kinhom</a> .                                                                                              |
| rmax                     | Maximum $r$ to evaluate $K(r)$ at. <code>rmax</code> is used to generate values for <code>r</code> , if omitted. If missing, a sensible default is chosen.                                                                                                  |
| kernel                   | Kernel type for smoothing of pcf.                                                                                                                                                                                                                           |
| bw                       | Kernel bandwidth for smoothing of pcf.                                                                                                                                                                                                                      |
| stoyan                   | Coefficient for Stoyan's bandwidth selection rule. See <a href="#">pcf.ppp</a> .                                                                                                                                                                            |
| normtol                  | A tolerance to use for <code>expectedPairs</code> or <code>expectedCrossPairs</code> when computing monte-carlo estimates of the normalizing factor $\gamma$ . Expressed as a maximum fractional standard error.                                            |
| ratio                    | If TRUE, assemble numerator and denominator of pcf estimator separately.                                                                                                                                                                                    |
| divisor                  | Whether to use the evaluation distance (" $r$ ") or the distance between points (" $d$ ") to normalize the contribution of each point pair.                                                                                                                 |
| analytical               | If TRUE, use Diggle-Jones weights                                                                                                                                                                                                                           |
| discrete.lambda          | If TRUE, and intensity function(s) are not supplied, estimate intensities by interpolating the values on a discrete lattice (using <a href="#">interp.im</a> and <a href="#">density.ppp</a> ), instead of exactly (using <a href="#">densityfun.ppp</a> ). |
| interpolate              | If TRUE, evaluate the <code>expectedCrossPairs</code> on a lattice and interpolate, rather than at the exact displacements observed in the pattern.                                                                                                         |

|                                |                                                                                                                                                                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>interpolate.fac</code>   | If <code>interpolate</code> , the lattice spacing will be <code>sigma/interpolate.fac</code> .                                                                                                                                                                                                                              |
| <code>leaveoneout</code>       | Use the leave-one-out estimator for $\gamma$ . See Shaw et al 2020 for details.                                                                                                                                                                                                                                             |
| <code>exp_prs</code>           | A function that returns values for $\gamma_{\text{iso}}(r)$ . If $\gamma$ is known explicitly, or the same calculation is being used for several point patterns, it can be much faster to compute it once and provide the function as <code>exp_prs</code> , since the computation of $\gamma$ is usually the slowest part. |
| <code>interpolate.maxdx</code> | Upper bound on allowable lattice spacing for interpolation.                                                                                                                                                                                                                                                                 |
| <code>dump</code>              | For debugging purposes, include computed values of $\gamma$ with the output, as at <code>trs</code> .                                                                                                                                                                                                                       |

### Value

The return value is an object of class `fv`, just as for `pcf` and `pcfinhom`. The object contains columns `r`, `theo`, and `global`, corresponding respectively to the argument `r`, the theoretical values of  $g(r)$  for a Poisson process, and  $g_{\text{global}}(r)$ .

### Author(s)

Thomas Shaw <shawtr@umich.edu>

### References

T Shaw, J Møller, R Waagepetersen. 2020. “Globally Intensity-Reweighted Estimators for  $K$ - and pair correlation functions”. arXiv:2004.00527 [stat.ME].

### See Also

[expectedPairs](#)

### Examples

```
rho <- funxy(function(x,y) 80*(1+x), owin())
X <- rpoispp(rho)
g <- pcfglobal(X)
#plot(g)
```



# Index

## \* package

globalKinhom-package, 2

density.ppp, 4, 5, 7

densityfun.ppp, 4, 5, 7

expectedCrossPairs (expectedPairs), 3

expectedCrossPairs\_iso (expectedPairs),  
3

expectedPairs, 3, 6, 8

expectedPairs\_iso (expectedPairs), 3

expectedPairs\_iso\_kernloo  
(expectedPairs), 3

expectedPairs\_kernloo (expectedPairs), 3

fv, 6, 8

globalKinhom (globalKinhom-package), 2

globalKinhom-package, 2

interp.im, 5, 7

Kcross.global (Kglobal), 5

Kcross.inhom, 2

Kest, 5–7

Kglobal, 3, 4, 5

Kinhom, 2, 5–7

pcf, 8

pcf.ppp, 7

pcfcross.global (pcfglobal), 6

pcfcross.inhom, 2

pcfglobal, 4, 6

pcfinhom, 2, 8

ppp, 4, 5, 7

spatstat.explore, 2, 3