Package 'pphotspot'

July 28, 2026

Version 0.1-2

Date 2026-07-18

Title Hotspot Detection of Point Events on a Linear Network

Encoding UTF-8

Maintainer Mari Myllymäki <mari.myllymaki@luke.fi>

Imports GET, parallel, sf, spatstat.geom, spatstat.Knet, spatstat.linnet, spatstat.random, stats

Suggests R.rsp, spatstat.explore, spatstat.model, testthat

Description Detection of hotspots of point events on a linear network as proposed by Mrkvička et al. (2025) <doi:10.2139/ssrn.5337003> using the R package 'GET', see Myllymäki and Mrkvička (2024) <doi:10.18637/jss.v111.i03>.

License GPL-3

RoxygenNote 7.3.3

URL https://github.com/myllym/pphotspot

BugReports https://github.com/myllym/pphotspot/issues

VignetteBuilder R.rsp

NeedsCompilation no

Author Mari Myllymäki [aut, cre] (ORCID: <https://orcid.org/0000-0002-2713-7088>), Michal Konopa [aut], Tomáš Mrkvička [aut], Mikko Kuronen [ctb]

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2026-07-28 16:10:02 UTC

Contents

`pphotspot-package` *Hotspot detection*

Description

The **pphotspot** package provides implementation of the detection of hotspots on a linear network as proposed by Mrkvička et al. (2025).

Key functions in GET

- `hotspots.poislpp` - Hotspot detection for Poisson point process
- `hotspots.MatClustlpp` - Hotspot detection for Matern point process
- `pois.lppm` - Fit a Poisson point process model to a point pattern dataset on a linear network
- `MatClust.lppm` - Fit a Matern cluster point process to a point pattern dataset on a linear network
- `rMatClustlpp` - Simulating the Matern cluster process on a linear network
- `roadcrash` - A data set of road crashes

Acknowledgements

Mikko Kuronen has helped to made the code faster.

Author(s)

Mari Myllymäki (mari.myllymaki@luke.fi, mari.j.myllymaki@gmail.com), Tomáš Mrkvička (mrkvicka.toma@gmail.com), and Michal Konopa (MKonopa@seznam.cz)

References

Mrkvička et al. (2025). Where are the Real Hotspots of Road Crashes? An Innovative Approach to the Road Crash Hotspot Detection by Filtering Out the Effect of Covariates. DOI: 10.2139/ssrn.5337003

`hotspots.MatClustlpp` *Hotspot detection for Matern point process - parallel version*

Description

See the hotspots vignette available by starting R, typing `library("GET")` and `vignette("GET")`.

Usage

```
hotspots.MatClustlpp(
  PP,
  formula,
  R,
  alpha,
  data,
  sigma = 250,
  nsim = 10000,
  ncores = 1L,
  ...
)
```

Arguments

`PP`	Input, a point pattern object (ppp) of spatstat.
`formula`	An R formula to estimate the first order model. This formula can contain objects of full size. `PP` should be on the left side of the formula.
`R`	Cluster radius parameter of the Matern cluster process.
`alpha`	Parameter related to mean number of points per cluster.
`data`	Data from where the formula takes objects. Must be acceptable by the function lppm of spatstat.linnet.
`sigma`	To be passed to density.lpp.
`nsim`	Number of simulations to be performed.
`ncores`	Number of cores used for computations. Default to 1. If NULL, all available cores are used.
`...`	Additional parameters to be passed to `fdr_envelope`.

Value

An object of type `fdr_envelope` and `global_envelope`, see further details in `global_envelope_test`.

References

Mrkvička et al. (2023). Hotspot detection on a linear network in the presence of covariates: A case study on road crash data. DOI: 10.2139/ssrn.4627591

`hotspots.poislpp` *Hotspot detection for Poisson point process - parallel version*

Description

See the hotspots vignette available by starting R, typing `library("GET")` and `vignette("GET")`.

Usage

```
hotspots.poislpp(
  PP,
  formula,
  data,
  sigma = 250,
  nsim = 10000,
  ncores = 1L,
  ...
)
```

Arguments

`PP`	The point pattern living in a network.
`formula`	A formula for the intensity.
`data`	Data from where the formula takes objects. Must be acceptable by the function lppm of spatstat.linnet.
`sigma`	To be passed to density.lpp.
`nsim`	Number of simulations to be performed.
`ncores`	Number of cores used for computations. Default to 1. If NULL, all available cores are used.
`...`	Additional parameters to be passed to `fdr_envelope`.

Value

An object of type `fdr_envelope` and `global_envelope`, see further details in `global_envelope_test`.

References

Mrkvička et al. (2023). Hotspot detection on a linear network in the presence of covariates: A case study on road crash data. DOI: 10.2139/ssrn.4627591

`MatClust.lppm` *Fit a Matern cluster point process to a point pattern dataset on a linear network*

Description

Fit a Matern cluster point process to a point pattern dataset on a linear network. This function is provided in **GET** to support `hotspots.MatClustlpp`. See the hotspots vignette available by starting R, typing `library("GET")` and `vignette("GET")`.

Usage

```
MatClust.lppm(
  PP,
  formula,
  subwin = NULL,
  valpha,
  vR,
  data,
  nsim = 10,
  ncores = 1L
)
```

Arguments

`PP`	Input, a point pattern object (ppp) of spatstat.
`formula`	An R formula to estimate the first order model. This formula can contain objects of full size. `PP` should be on the left side of the formula.
`subwin`	A part of the observation window of `PP` to be used for estimating the second order structure. NULL means that the full point pattern is used. Typically this is feasible (not too time consuming).
`valpha`	A vector of parameter values for the parameter alpha of the Matern cluster process.
`vR`	A vector of parameter values for the parameter R of the Matern cluster process.
`data`	Data from where the formula takes objects. Must be acceptable by the function lppm of spatstat.linnet.
`nsim`	The number of simulated Matern cluster point patterns for evaluating the K-function for any alpha and R values.
`ncores`	Number of cores used for computations. Default to 1. If NULL, all available cores are used.

Details

The function MatClust.lppm, can be used to estimate the Matern cluster point pattern with inhomogeneous cluster centers on linear network. This function provides the same outputs as the pois.lppm and further estimated parameters alpha and R. The secondorder provides again the diagnostics for checking if the clustered model is appropriate. The sample K-function must be close to the K-function of the estimated model (green line). If it is not the case the searching grid for parameters alpha and R that is input in the function must be manipulated to get the a closer result. If the estimated model is adequate one can proceed to the hotspot detection with the use of the function hotspots.MatClustlpp. Remark here, that for the estimation of the second order structure a smaller data can be used than for the estimation of the first order structure in order to save the computation time, since the second order is a local characteristics. This smaller window can be specified by subwin. Then the full point pattern will be used for estimation of first order intensity and the pattern in subwindow will be used for estimating second order characteristic. The input parameters are the same as in pois.lppm. Furthermore, valpha, i.e., vector of proposed alphas which should be considered in the optimization, vR, i.e., vector of proposed values for R which should be considered in the optimization, must be provided. The user can also specify how many cores should be used in the computation by parameter ncores.

Value

A list of

- EIP = the predicted point process intensity using the firstordermodel
- alpha = the estimated value of the parameter alpha of the Matern cluster process, see rMatClustlpp
- R = the estimated value of the parameter R of the Matern cluster process, see rMatClustlpp
- Contrast = the minimum value of the contrast for the estimated alpha and R; contrast is sqrt(sum((estK - meanK)^2)) where estK is the K-function estimated from the data pattern and meanK is the mean estimated K function from nsim simulations from the Matern cluster process with the estimated alpha and R
- firstordermodel = the model fitted for the intensity using the full pattern PPfull (provided to the MatClust.lppm function in the argument PP)
- secondorder = estK, i.e. the K-function estimated from the data pattern
- MCsecondorder = meanK, i.e. the mean estimated K function from nsim simulations from the Matern cluster process with the estimated alpha and R

---

pois.lppm *Fit a Poisson point process model to a point pattern dataset on a linear network*

Description

Fit a Poisson point process model to a point pattern dataset on a linear network. This function is provided in **GET** to support hotspots.poislpp and hotspots.MatClustlpp. See the hotspots vignette available by starting R, typing library("GET") and vignette("GET").

Usage

```
pois.lppm(PP, formula, data, subwin = NULL, r_max = NULL)
```

Arguments

PP	Input, a point pattern object (ppp) of spatstat.
formula	An R formula to estimate the first order model. This formula can contain objects of full size. PP should be on the left side of the formula.
data	Data from where the formula takes objects. Must be acceptable by the function lppm of spatstat.linnet.
subwin	A part of the observation window of PP to be used for estimating the second order structure. NULL means that the full point pattern is used. Typically this is feasible (not too time consuming).
r_max	The maximum distance on which the K-function is evaluated. Default is computed as $\sqrt{A}/10$ where A is the area of the window of observation of X.

Details

The function pois.lppm, can be used to estimate the inhomogeneous Poisson point process model on linear network. This function provides the firstordermodel, i.e. the regression model of dependence of crashes on the spatial covariates, EIP, i.e. estimated inhomogeneous intensity from the data and secondorder, i.e. estimation of the inhomogeneous K-function. The plot of the secondorder provides diagnostics, if the model is adequate for the data. If the estimated K-function lies close to the theoretical line, the data does not report any clustering, and the function hotspots.poislpp can be used for final hotspots detection. If the estimated K-function does not lie close to the theoretical line, and it is above, the data report clustering, and the a clustered point pattern model must be fitted to the data and hotspots detected using this clustered model instead.

Value

A list of three components:

- EIP = the predicted point process intensity
- firstordermodel = the model fitted for the intensity
- secondorder = the inhomogeneous K-function estimated from the given point pattern PP using the estimated intensity

---

rMatClustlpp *Simulating the Matern cluster process on a linear network*

Description

Simulating the Matern clusters with parameters alpha and R given the (x,y)- coordinates of the parent points. This function is provided in **GET** to support hotspots.MatClustlpp. See the hotspots vignette available by starting R, typing library("GET") and vignette("GET").

Usage

```
rMatClustlpp(Centers, R, alpha, LL, check_vol = FALSE)
```

Arguments

Centers	The (x,y)-coordinates of parent points
R	Cluster radius parameter of the Matern cluster process.
alpha	Parameter related to mean number of points per cluster.
LL	The linear network on which the point pattern should be simulated.
check_vol	Logical. TRUE for checking if the ball producing the cluster has any intersection with linear network.

Value

A point pattern on linear network, lpp object of spatstat.

Examples

```
# example code
if(require(spatstat.geom, quietly=TRUE) & require(spatstat.linnet, quietly=TRUE)) {
data("roadcrash")
win <- owin(xrange = roadcrash$xrange,
            yrange = roadcrash$yrange)
X <- ppp(x = roadcrash$x, y = roadcrash$y, window = win)
Vertices.pp <- ppp(x = roadcrash$Vertices.x,
                   y = roadcrash$Vertices.y,
                   window=win)
L <- linnet(vertices=Vertices.pp,
            edges = roadcrash$Edges)
PPfull <- lpp(X, L)
subwin <- owin(c(-760000, -740000), c(-1160000, -1140000))
PPsub <- PPfull[, subwin]
Centers <- rpoislpp(0.0005, L=PPsub[['domain']])
XX <- rMatClustlpp(Centers, 500, 5, PPsub[['domain']])
plot(Centers, col=1, pch=16)
points(XX, col=2)
}
```

---

roadcrash *Road crashes*

Description

Road crashes

Usage

```
data("roadcrash")
```

Format

A `list` with

- x = x-coordinates of road accidents
- y = y-coordinates of road accidents
- xrange = x coordinate limits of enclosing box (-774936.86,-727048.86)
- yrange = y coordinate limits of enclosing box (-1201599.83,-1125679.83)
- Vertices.x = x-coordinates of vertices of the linear network
- Vertices.y = y-coordinates of vertices of the linear network
- Edges = a 2 column matrix giving the ID (index) of the origin and destination vectices (in Vertices.x and Vertices.y)
- Traffic = matrix of traffic volume
- ForestDensity = matrix of forest density
- BuildingDensity = matrix of building density

Details

Mrkvička et al. (2023) worked with the database of road crashes reported to the Police in the Czech Republic from 1 January 2016 to 31 December 2020. The data available here is a subpattern of this full data set, included here with the permission of the Police in the Czech Republic. The full data is published as open data, see https://policie.gov.cz/clanek/statistika-nehodovosti.aspx and https://nehody.cdv.cz/statistics.php. The subpattern 7700 crashes lying on a linear network with 269 vertices and 354 lines. Further average traffic volume (number of vehicles per 24 hours), forest density and building density in the cell are available in the region of the linear network.

References

Mrkvička, T., Kraft, S., Blažek, V. and Myllymäki, M. (2023) Hotspots detection on a linear network with presence of covariates: a case study on road crash data.

Examples

```
if(require("spatstat.geom", quietly = TRUE) & require("spatstat.linnet", quietly = TRUE)) {
  data("roadcrash")
  win <- owin(xrange = roadcrash$xrange,
              yrange = roadcrash$yrange)
  X <- ppp(x = roadcrash$x, y = roadcrash$y, window = win)
  Vertices.pp <- ppp(x = roadcrash$Vertices.x,
                     y = roadcrash$Vertices.y,
                     window=win)
  L <- linnet(vertices=Vertices.pp,
              edges = roadcrash$Edges)
  PP <- lpp(X, L)
  z1 <- im(roadcrash$Traffic,
           xrange = roadcrash$xrange,
           yrange = roadcrash$yrange)
```

```
  z2 <- im(roadcrash$ForestDensity,
           xrange = roadcrash$xrange,
           yrange = roadcrash$yrange)
  z3 <- im(roadcrash$BuildingDensity,
           xrange = roadcrash$xrange,
           yrange = roadcrash$yrange)
}
```

Index