

Package ‘stpp’

April 6, 2020

Version 2.0-4

Title Space-Time Point Pattern Simulation, Visualisation and Analysis

Date 2020-04-01

Maintainer Edith Gabriel <edith.gabriel@inrae.fr>

Depends R (>= 3.3.0), grDevices, graphics, rpanel, splancs

Imports KernSmooth, ggplot2, gridExtra, plot3D, rgl, spatstat, stats

Suggests knitr(>= 1.11), rmarkdown(>= 0.8.1)

Description Many of the models encountered in applications of point process methods to the study of spatio-temporal phenomena are covered in 'stpp'. This package provides statistical tools for analyzing the global and local second-order properties of spatio-temporal point processes, including estimators of the space-time inhomogeneous K-function and pair correlation function. It also includes tools to get static and dynamic display of spatio-temporal point patterns. See Gabriel et al (2013) <doi:10.18637/jss.v053.i02>.

License GPL-3

LazyLoad yes

NeedsCompilation yes

RoxygenNote 6.0.1

BuildVignettes True

VignetteBuilder knitr

BugReports <https://github.com/stpp-GitHub-community>

Author Edith Gabriel [aut, cre],
Peter J Diggle [aut],
Barry Rowlingson [aut],
Francisco J Rodriguez-Cortes [aut]

Repository CRAN

Date/Publication 2020-04-06 14:10:02 UTC

R topics documented:

animation	2
as.3dpoints	3
ASTIKhat	4
fmd	5
gsp	6
gte	7
is.3dpoints	9
KLISTAhat	10
LISTAhat	11
northcumbria	13
PCFhat	14
plot.stpp	17
plotK	18
plotPCF	19
rinfec	20
rinter	22
rlgcp	24
rpcp	27
rpp	29
sim.stpp	31
stan	32
stdcpp	33
sthpcpp	35
STIKhat	37
stpp	40
Index	43

 animation

Space-time data animation

Description

Provide an animation of spatio-temporal point patterns.

Usage

```
animation(xyt, s.region, t.region, runtime=1, incident="red",
prevalent="pink3", pch=19, cex=0.5, plot.s.region=TRUE,
scales=TRUE, border.frac=0.05, add=FALSE)
```

Arguments

<code>xyt</code>	Data-matrix containing the (x, y, t) -coordinates.
<code>s.region</code>	Two-column matrix specifying polygonal region containing all data-locations <code>xyt[,1:2]</code> . If missing, <code>s.region</code> is the bounding box of <code>xyt[,1:2]</code> .
<code>t.region</code>	Interval containing all data-times <code>xyt[,3]</code> . If missing, <code>t.region</code> is defined by the range of <code>xyt[,3]</code> .
<code>runtime</code>	Approximate running time of animation, in seconds, but it is longer than expected. Can also be NULL.
<code>incident</code>	Colour in which incident point <code>xyt[i,1:2]</code> is plotted at time <code>xyt[i,3]</code> .
<code>prevalent</code>	Colour to which prevalent point <code>xyt[i,1:2]</code> fades at time <code>xyt[i+1,3]</code> .
<code>pch</code>	Plotting symbol used for each point.
<code>cex</code>	Magnification of plotting symbol relative to standard size.
<code>plot.s.region</code>	If TRUE, plot <code>s.region</code> as polygon.
<code>scales</code>	If TRUE, plot X and Y axes with scales.
<code>border.frac</code>	Extent of border of plotting region surrounding <code>s.region</code> , as fraction of ranges of X and Y.
<code>add</code>	If TRUE, add the animation to an existing plot.

Value

None

Author(s)

Peter J Diggle, Edith Gabriel <edith.gabriel@inrae.fr>.

as.3dpoints*Create data in spatio-temporal point format*

Description

Create data in spatio-temporal point format.

Usage`as.3dpoints(...)`**Arguments**

`...` Any object(s), such as x , y and t vectors of the same length, or a list or data frame containing x , y and t vectors. Valid options for `...` are: a points object: returns it unaltered; a list with x , y and t elements of the same length: returns a points object with the x , y and t elements as the coordinates of the points; three vectors of equal length: returns a points object with the first vector as the x coordinates, the second vector as the y -coordinates and the third vector as the t -coordinates.

Value

The output is an object of the class `stpp.as3dpoints` tries to return the argument(s) as a spatio-temporal points object.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>, Peter Diggle, Barry Rowlingson.

 ASTIKhat

Anisotropic Space-Time Inhomogeneous K-function

Description

Compute an estimation of the Anisotropic Space-Time inhomogeneous K -function.

Usage

```
ASTIKhat(xyt, s.region, t.region, lambda, dist, times, ang,
         correction = "border")
```

Arguments

<code>xyt</code>	Coordinates and times (x, y, t) of the point pattern.
<code>s.region</code>	Two-column matrix specifying polygonal region containing all data locations. If <code>s.region</code> is missing, the bounding box of <code>xyt[, 1:2]</code> is considered.
<code>t.region</code>	Vector containing the minimum and maximum values of the time interval. If <code>t.region</code> is missing, the range of <code>xyt[, 3]</code> is considered.
<code>dist</code>	Vector of distances u at which $\widehat{K}_\phi(r, t)$ is computed. If missing, the maximum of <code>dist</code> is given by $\min(S_x, S_y)/4$, where S_x and S_y represent the maximum width and height of the bounding box of <code>s.region</code> .
<code>times</code>	Vector of times v at which $\widehat{K}_\phi(r, t)$ is computed. If missing, the maximum of <code>times</code> is given by $(T_{\max} - T_{\min})/4$, where T_{\min} and T_{\max} are the minimum and maximum of the time interval T .
<code>lambda</code>	Vector of values of the space-time intensity function evaluated at the points (x, y, t) in $S \times T$. If <code>lambda</code> is missing, the estimate of the anisotropic space-time K -function is computed as for the homogeneous case, i.e. considering $n/ S \times T $ as an estimate of the space-time intensity.
<code>ang</code>	Angle in radians at which $\widehat{K}_\phi(r, t)$ is computed. The argument <code>ang=2*pi</code> by default.
<code>correction</code>	A character vector specifying the edge correction(s) to be applied among "border", "modified.border", "translate" and "none" (see STIKhat). The default is "border".

Value

A list containing:

AKhat	ndist x ntimes matrix containing values of $\widehat{K}_\phi(u, t)$.
dist, times	Parameters passed in argument.
correction	The name(s) of the edge correction method(s) passed in argument.

Author(s)

Francisco J. Rodriguez-Cortes <frrodriguez@unal.edu.co>

References

Illian, J. B., Penttinen, A., Stoyan, H. and Stoyan, D. (2008). Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley and Sons, London.

Gonzalez, J. A., Rodriguez-Cortes, F. J., Cronie, O., Mateu, J. (2016). Spatio-temporal point process statistics: a review. Spatial Statistics. Accepted.

Ohser, J. and D. Stoyan (1981). On the second-order and orientation analysis of planar stationary point processes. Biometrical Journal 23, 523-533.

fmd

2001 food-and-mouth epidemic, north Cumbria (UK)

Description

This data set gives the spatial locations and reported times of food-and-mouth disease in north Cumbria (UK), 2001. It is of no scientific value, as it deliberately excludes confidential information on farms at risk in the study-region. It is included in the package purely as an illustrative example.

Usage

`data(fmd)`

Format

A matrix containing (x, y, t) coordinates of the 648 observations.

References

Diggle, P., Rowlingson, B. and Su, T. (2005). Point process methodology for on-line spatio-temporal disease surveillance. Environmetrics, 16, 423–34.

See Also

[northcumbria](#) for boundaries of the county of north Cumbria.

`gsp` *Spatial mark variogram function*

Description

Computes an estimator of the spatial mark variogram function.

Usage

```
gsp(xyt, s.region, s.lambda, ds, ks="epanech", hs, correction="none", approach="simplified")
```

Arguments

<code>xyt</code>	Spatial coordinates and times (x, y, t) of the point pattern.
<code>s.region</code>	Two-column matrix specifying polygonal region containing all data locations.
<code>s.lambda</code>	Vector of values of the spatial intensity function evaluated at the points (x, y) in W . If <code>s.lambda</code> is missing, the estimate of the spatial mark correlation function is computed as for the homogeneous case, i.e. considering $n/ W $ as an estimate of the spatial intensity under the parameter <code>approach="standardised"</code> .
<code>ds</code>	A vector of distances u at which <code>gsp(u)</code> is computed.
<code>ks</code>	A kernel function for the spatial distances. The default is the "epanech" kernel. It can also be "box" for the uniform kernel, or "biweight".
<code>hs</code>	A bandwidth of the kernel function <code>ks</code> .
<code>correction</code>	A character vector specifying the edge-correction(s) to be applied among "isotropic", "border", "modified.border", "translate", "setcovf" and "none". The default is "none".
<code>approach</code>	A character vector specifying the approach to use for the estimation to be applied among "simplified" or "standardised". If <code>approach</code> is missing, "simplified" is considered by default.

Details

By default, this command calculates an estimate of the spatial mark variogram function $\gamma_{[sp]}(r)$ for a spatio-temporal point pattern.

Value

<code>egsp</code>	A vector containing the values of $\gamma_{sp}(r)$ estimated
<code>ds</code>	If <code>ds</code> is missing, a vector of distances u at which <code>gsp(u)</code> is computed from 0 to until quarter of the maximum distance between the points in the pattern.
<code>kernel</code>	A vector of names and bandwidth of the spatial kernel.
<code>gsptheo</code>	Value under the Poisson case is calculated considering $\tau = \max(xyt[, 3]) - \min(xyt[, 3])$.

Author(s)

Francisco J. Rodriguez Cortes <frrodriguez@unal.edu.co> <https://fjrodriguezcortes.wordpress.com>

References

Baddeley, A., Rubak, E., Turner, R. (2015). Spatial Point Patterns: Methodology and Applications with R. CRC Press, Boca Raton.

Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). Stochastic Geometry and its Applications. John Wiley & Sons.

Gabriel, E., Rowlingson, B., Diggle P J. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software. 53, 1-29.

Illian, J B., Penttinen, A., Stoyan, H. and Stoyan, D. (2008). Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley and Sons, London.

Stoyan, D., Rodriguez-Cortes, F. J., Mateu, J., and Gille, W. (2017). Mark variograms for spatio-temporal point processes. Spatial Statistics. 20, 125-147.

Examples

```
## Not run:
#####

# A realisation of spatio-temporal homogeneous Poisson point processes
hpp <- rpp(lambda = 100, replace = FALSE)$xyt

# R plot
plot(hpp)

# This function provides an kernel estimator of the spatial mark variogram function
out <- gsp(hpp)

# R plot - Spatial mark variogram function
par(mfrow=c(1,1))
xl <- c(0,0.25)
yl <- c(0,max(out$gspttheo,out$egsp))
plot(out$ds,out$egsp,type="l",xlab="r = distance",ylab=expression(gamma[sp](r)),
      xlim=xl,ylim=yl,col=1,cex.lab=1.5,cex.axis=1.5)
lines(out$ds,rep(out$gspttheo,length(out$ds)),col=11)

## End(Not run)
```

 gte

Temporal mark variogram function

Description

Computes an estimator of the temporal mark variogram function.

Usage

```
gte(xyt, t.region, t.lambda, dt, kt="epanech", ht, correction="none", approach="simplified")
```

Arguments

xyt	Spatial coordinates and times (x, y, t) of the point pattern.
t.region	Vector containing the minimum and maximum values of the time interval.
t.lambda	Vector of values of the temporal intensity function evaluated at the points t in T . If <code>t.lambda</code> is missing, the estimate of the temporal mark correlation function is computed as for the homogeneous case, i.e. considering $n/ T $ as an estimate of the temporal intensity under the parameter <code>approach="standardised"</code> .
dt	A vector of times v at which <code>gte(v)</code> is computed.
kt	A kernel function for the temporal distances. The default is the "epanech" kernel. It can also be "box" for the uniform kernel, or "biweight".
ht	A bandwidth of the kernel function <code>kt</code> .
correction	A character vector specifying the edge-correction(s) to be applied among "isotropic", "border", "modified.border", "translate", "setcovf" and "none". The default is "none".
approach	A character vector specifying the approach to use for the estimation to be applied among "simplified" or "standardised". If <code>approach</code> is missing, "simplified" is considered by default.

Details

By default, this command calculates an estimate of the temporal mark variogram function $\gamma_{[te]}(t)$ for a spatio-temporal point pattern.

Value

egte	A vector containing the values of $\gamma_{[te]}(v)$ estimated.
dt	Parameter passed in argument. If <code>dt</code> is missing, a vector of temporal distances v at which <code>gte(v)</code> is computed from 0 to until quarter of the maximum distance between the times in the temporal pattern.
kernel	Parameters passed in argument. A vector of names and bandwidth of the spatial kernel.
gtetheo	Value under the Poisson case is calculated considering the side lengths of the bounding box of <code>xyt[, 1:2]</code> .

Author(s)

Francisco J. Rodriguez Cortes <frrodriguez@unal.edu.co> <https://fjrodriguezcortes.wordpress.com>

References

- Baddeley, A., Rubak, E., Turner, R. (2015). Spatial Point Patterns: Methodology and Applications with R. CRC Press, Boca Raton.
- Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). Stochastic Geometry and its Applications. John Wiley & Sons.
- Gabriel, E., Rowlingson, B., Diggle P J. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software 53, 1-29.
- Illian, J B., Penttinen, A., Stoyan, H. and Stoyan, D. (2008). Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley and Sons, London.
- Stoyan, D., Rodriguez-Cortes, F. J., Mateu, J., and Gille, W. (2017). Mark variograms for spatio-temporal point processes. Spatial Statistics. 20, 125-147.

Examples

```
## Not run:
#####

# A realisation of spatio-temporal homogeneous Poisson point processes
hpp <- rpp(lambda = 100, replace = FALSE)$xyt

# R plot
plot(hpp)

# This function provides an kernel estimator of the temporal mark variograma function
out <- gte(hpp)

# R plot - Temporal mark variogram function
par(mfrow=c(1,1))
xl <- c(0,0.25)
yl <- c(0,max(out$egte,out$gtetheo))
plot(out$dt,out$egte,type="l",xlab="t = time",ylab=expression(gamma[te](t)),
      xlim=xl,ylim=yl,col=1,cex.lab=1.5,cex.axis=1.5)
lines(out$dt,rep(out$gtetheo,length(out$dt)),col=11)

## End(Not run)
```

is.3dpoints

Spatio-temporal point objects

Description

Tests for data in spatio-temporal point format.

Usage

```
is.3dpoints(x)
```

Arguments

x any object.

Value

is.3dpoints returns TRUE if x is a spatio-temporal points object, FALSE otherwise.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>, Peter Diggle, Barry Rowlingson.

KLISTAhat

Estimation of the Space-Time Inhomogeneous K LISTA functions

Description

Compute an estimate of the space-time K LISTA functions.

Usage

KLISTAhat(xyt, s.region, t.region, dist, times, lambda, correction = "isotropic")

Arguments

xyt	Coordinates and times (x, y, t) of the point pattern.
s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the bounding box of xyt[, 1:2] is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the range of xyt[, 3] is considered.
dist	Vector of distances u at which $K^{(i)}(u, v)$ is computed. If missing, the maximum of dist is given by $\min(S_x, S_y)/4$, where S_x and S_y represent the maximum width and height of the bounding box of s.region.
times	Vector of times v at which $K^{(i)}(u, v)$ is computed. If missing, the maximum of times is given by $(T_{\max} - T_{\min})/4$, where T_{\min} and T_{\max} are the minimum and maximum of the time interval T .
lambda	Vector of values of the space-time intensity function evaluated at the points (x, y, t) in $S \times T$. If lambda is missing, the estimate of the space-time pair correlation function is computed as for the homogeneous case, i.e. considering $(n - 1)/ S \times T $ as an estimate of the space-time intensity.
correction	A character vector specifying the edge correction(s) to be applied among "isotropic", "border", "modified.border", "translate" and "none" (see PCFhat). The default is "isotropic".

Details

An individual product density LISTA functions $K^{(i)}(.,.)$ should reveal the extent of the contribution of the event (u_i, t_i) to the global estimator of the K-function $K(.,.)$, and may provide a further description of structure in the data (e.g., determining events with similar local structure through dissimilarity measures of the individual LISTA functions), for more details see Siino et al. (2019).

Value

A list containing:

<code>list.KLISTA</code>	A list containing the values of the estimation of $K^{(i)}(r, t)$ for each one of n points of the point pattern by matrixs.
<code>klistatheo</code>	<code>ndist x ntimes</code> matrix containing theoretical values for a Poisson process.
<code>dist, times</code>	Parameters passed in argument.
<code>correction</code>	The name(s) of the edge correction method(s) passed in argument.

Author(s)

Francisco J. Rodriguez-Cortes <frodriquezc@unal.edu.co>

References

- Baddeley, A. and Turner, J. (2005). `spatstat`: An R Package for Analyzing Spatial Point Pattens. *Journal of Statistical Software* 12, 1-42.
- Cressie, N. and Collins, L. B. (2001). Analysis of spatial point patterns using bundles of product density LISA functions. *Journal of Agricultural, Biological, and Environmental Statistics* 6, 118-135.
- Cressie, N. and Collins, L. B. (2001). Patterns in spatial point locations: Local indicators of spatial association in a minefield with clutter Naval Research Logistics (NRL), John Wiley & Sons, Inc. 48, 333-347.
- Siino, M., Adelfio, G., Mateu, J. and Rodriguez-Cortes, F. J. (2019). Some properties of weighted local second-order statistics for spatio-temporal point process. Submitted.
- Stoyan, D. and Stoyan, H. (1994). *Fractals, random shapes, and point fields: methods of geometrical statistics*. Chichester: Wiley.

LISTAhat

Estimation of the Space-Time Inhomogeneous Pair Correlation LISTA functions

Description

Compute an estimate of the space-time pair correlarion LISTA functions.

Usage

```
LISTAhat(xyt, s.region, t.region, dist, times, lambda,
         ks = "box", hs, kt = "box", ht, correction = "isotropic")
```

Arguments

xyt	Coordinates and times (x, y, t) of the point pattern.
s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the bounding box of <code>xyt[, 1:2]</code> is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the range of <code>xyt[, 3]</code> is considered.
dist	Vector of distances u at which $g^{(i)}(u, v)$ is computed. If missing, the maximum of dist is given by $\min(S_x, S_y)/4$, where S_x and S_y represent the maximum width and height of the bounding box of s.region.
times	Vector of times v at which $g^{(i)}(u, v)$ is computed. If missing, the maximum of times is given by $(T_{\max} - T_{\min})/4$, where T_{\min} and T_{\max} are the minimum and maximum of the time interval T .
lambda	Vector of values of the space-time intensity function evaluated at the points (x, y, t) in $S \times T$. If lambda is missing, the estimate of the space-time pair correlation function is computed as for the homogeneous case, i.e. considering $(n - 1)/ S \times T $ as an estimate of the space-time intensity.
ks	Kernel function for the spatial distances. Default is the "box" kernel. Can also be "epanech" for the Epanechnikov kernel or "gaussian" or "biweight".
hs	Bandwidth of the kernel function ks.
kt	Kernel function for the temporal distances. Default is the "box" kernel. Can also be "epanech" for the Epanechnikov kernel or "gaussian" or "biweight".
ht	Bandwidth of the kernel function kt.
correction	A character vector specifying the edge correction(s) to be applied among "isotropic", "border", "modified.border", "translate" and "none" (see PCFhat). The default is "isotropic".

Details

An individual product density LISTA functions $g^{(i)}(\cdot, \cdot)$ should reveal the extent of the contribution of the event (u_i, t_i) to the global estimator of the pair correlation function $g(\cdot, \cdot)$, and may provide a further description of structure in the data (e.g., determining events with similar local structure through dissimilarity measures of the individual LISTA functions), for more details see Siino et al. (2017).

Value

A list containing:

list.LISTA	A list containing the values of the estimation of $g^{(i)}(r, t)$ for each one of n points of the point pattern by matrixs.
listatheo	ndist x ntimes matrix containing theoretical values for a Poisson process.

dist, times	Parameters passed in argument.
kernel	Vector of names and bandwidths of the spatial and temporal kernels.
correction	The name(s) of the edge correction method(s) passed in argument.

Author(s)

Francisco J. Rodriguez-Cortes <frrodriguezc@unal.edu.co>

References

Baddeley, A. and Turner, J. (2005). spatstat: An R Package for Analyzing Spatial Point Patterns. Journal of Statistical Software 12, 1-42.

Cressie, N. and Collins, L. B. (2001). Analysis of spatial point patterns using bundles of product density LISA functions. Journal of Agricultural, Biological, and Environmental Statistics 6, 118-135.

Cressie, N. and Collins, L. B. (2001). Patterns in spatial point locations: Local indicators of spatial association in a minefield with clutter Naval Research Logistics (NRL), John Wiley & Sons, Inc. 48, 333-347.

Siino, M., Rodriguez-Cortes, F. J., Mateu, J. and Adelfio, G. (2017). Testing for local structure in spatio-temporal point pattern data. Environmetrics. DOI: 10.1002/env.2463.

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

northcumbria	<i>Polygon boundary of north Cumbria</i>
--------------	--

Description

This data set gives the boundary of the county of north Cumbria (UK).

Usage

```
data(northcumbria)
```

Format

A matrix containing (x, y) coordinates of the boundary.

See Also

[fmd](#) for the space-time pattern of food-and-mouth disease in this county in 2001.

PCFhat	<i>Estimation of the Space-Time Inhomogeneous Pair Correlation function</i>
--------	---

Description

Compute an estimate of the space-time pair correlation function.

Usage

```
PCFhat(xyt, s.region, t.region, dist, times, lambda,
       ks="box", hs, kt="box", ht, correction = "isotropic")
```

Arguments

xyt	Coordinates and times (x, y, t) of the point pattern.
s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the bounding box of xyt[, 1:2] is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the range of xyt[, 3] is considered.
dist	Vector of distances u at which $g(u, v)$ is computed. If missing, the maximum of dist is given by $\min(S_x, S_y)/4$, where S_x and S_y represent the maximum width and height of the bounding box of s.region.
times	Vector of times v at which $g(u, v)$ is computed. If missing, the maximum of times is given by $(T_{\max} - T_{\min})/4$, where T_{\min} and T_{\max} are the minimum and maximum of the time interval T .
lambda	Vector of values of the space-time intensity function evaluated at the points (x, y, t) in $S \times T$. If lambda is missing, the estimate of the space-time pair correlation function is computed as for the homogeneous case, i.e. considering $n/ S \times T $ as an estimate of the space-time intensity.
ks	Kernel function for the spatial distances. Default is the "box" kernel. Can also be "epanech" for the Epanechnikov kernel or "gaussian" or "biweight".
hs	Bandwidth of the kernel function ks.
kt	Kernel function for the temporal distances. Default is the "box" kernel. Can also be "epanech" for the Epanechnikov kernel or "gaussian" or "biweight".
ht	Bandwidth of the kernel function kt.
correction	A character vector specifying the edge correction(s) to be applied among "isotropic", "border", "modified.border", "translate" and "none" (see Details). The default is "isotropic".

Details

An approximately unbiased estimator for the space-time pair correlation function, based on data giving the locations of events $x_i : i = 1, \dots, n$ on a spatio-temporal region $S \times T$, where S is an arbitrary polygon and T a time interval:

$$\hat{g}(u, v) = \frac{1}{4\pi u} \sum_{i=1}^n \sum_{j \neq i} \frac{1}{w_{ij}} \frac{k_s(u - \|s_i - s_j\|)k_t(v - |t_i - t_j|)}{\lambda(x_i)\lambda(x_j)},$$

where $\lambda(x_i)$ is the intensity at $x_i = (s_i, t_i)$ and w_{ij} is an edge correction factor to deal with spatial-temporal edge effects. The edge correction methods implemented are:

isotropic: $w_{ij} = |S \times T| w_{ij}^{(t)} w_{ij}^{(s)}$, where the temporal edge correction factor $w_{ij}^{(t)} = 1$ if both ends of the interval of length $2|t_i - t_j|$ centred at t_i lie within T and $w_{ij}^{(t)} = 1/2$ otherwise and $w_{ij}^{(s)}$ is the proportion of the circumference of a circle centred at the location s_i with radius $\|s_i - s_j\|$ lying in S (also called Ripley's edge correction factor).

border: $w_{ij} = \frac{\sum_{j=1}^n \mathbf{1}_{\{d(s_j, S) > u ; d(t_j, T) > v\}} / \lambda(x_j)}{\mathbf{1}_{\{d(s_i, S) > u ; d(t_i, T) > v\}}}$, where $d(s_i, S)$ denotes the distance between s_i and the boundary of S and $d(t_i, T)$ the distance between t_i and the boundary of T .

modified.border: $w_{ij} = \frac{|S_{\ominus u}| \times |T_{\ominus v}|}{\mathbf{1}_{\{d(s_i, S) > u ; d(t_i, T) > v\}}}$, where $S_{\ominus u}$ and $T_{\ominus v}$ are the eroded spatial and temporal region respectively, obtained by trimming off a margin of width u and v from the border of the original region.

translate: $w_{ij} = |S \cap S_{s_i - s_j}| \times |T \cap T_{t_i - t_j}|$, where $S_{s_i - s_j}$ and $T_{t_i - t_j}$ are the translated spatial and temporal regions.

none: No edge correction is performed and $w_{ij} = |S \times T|$.

$k_s()$ and $k_t()$ denotes kernel functions with bandwidth h_s and h_t . Experience with pair correlation function estimation recommends box kernels (the default), see Illian et al. (2008). Epanechnikov, Gaussian and biweight kernels are also implemented. Whatever the kernel function, if the bandwidth is missing, a value is obtain from the function `dpik` of the package `KernSmooth`. Note that the bandwidths play an important role and their choice is crucial in the quality of the estimators as they heavily influence their variance.

Value

A list containing:

<code>pcf</code>	<code>ndist x ntimes</code> matrix containing values of $\hat{g}(u, v)$.
<code>pcftheo</code>	<code>ndist x ntimes</code> matrix containing theoretical values for a Poisson process.
<code>dist, times</code>	Parameters passed in argument.
<code>kernel</code>	A vector of names and bandwidths of the spatial and temporal kernels.
<code>correction</code>	The name(s) of the edge correction method(s) passed in argument.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>

References

- Baddeley, A., Rubak, E., Turner, R., (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, Boca Raton.
- Gabriel E., Diggle P. (2009). Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica*, 63, 43–51.
- Gabriel E., Rowlingson B., Diggle P. (2013). *stpp*: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. *Journal of Statistical Software*, 53(2), 1–29.
- Gabriel E. (2014). Estimating second-order characteristics of inhomogeneous spatio-temporal point processes: influence of edge correction methods and intensity estimates. *Methodology and computing in Applied Probability*, 16(2), 411–431.
- Illian JB, Penttinen A, Stoyan H and Stoyan, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons, London.

Examples

```
# First example

data(fmd)
data(northcumbria)
FMD<-as.3dpoints(fmd[,1]/1000,fmd[,2]/1000,fmd[,3])
Northcumbria=northcumbria/1000

# estimation of the temporal intensity
Mt<-density(FMD[,3],n=1000)
mut<-Mt$y[findInterval(FMD[,3],Mt$x)]*dim(FMD)[1]

# estimation of the spatial intensity
h<-mse2d(as.points(FMD[,1:2]), Northcumbria, nsmse=50, range=4)
h<-h$h[which.min(h$mse)]
Ms<-kernel2d(as.points(FMD[,1:2]), Northcumbria, h, nx=500, ny=500)
atx<-findInterval(x=FMD[,1],vec=Ms$x)
aty<-findInterval(x=FMD[,2],vec=Ms$y)
mhat<-NULL
for(i in 1:length(atx)) mhat<-c(mhat,Ms$z[atx[i],aty[i]])

# estimation of the pair correlation function
g1 <- PCFhat(xyt=FMD, dist=1:15, times=1:15, lambda=mhat*mut/dim(FMD)[1],
  s.region=northcumbria/1000,t.region=c(1,200))

# plotting the estimation

plotPCF(g1)
plotPCF(g1,type="persp",theta=-65,phi=35)

# Second example

xyt=rpp(lambda=200)
g2=PCFhat(xyt$xyt,dist=seq(0,0.16,by=0.02),
```

```
times=seq(0,0.16,by=0.02),correction=c("border","translate"))
plotPCF(g2,type="contour",which="border")
```

plot.stpp

Plot for spatio-temporal point objects

Description

This function plot either *xy*-locations and cumulative distribution of the times, or a space-time 3D scatter, or the time-mark and space-mark of the spatio-temporal point pattern, through arguments *style* and *type*.

It can also plot *xy*-locations with time treated as a quantitative mark attached to each location, as in the previous version of the function, through argument *mark* (see *stpp* version < 2.0.0).

Usage

```
## S3 method for class 'stpp'
plot(x, s.region=NULL, t.region=NULL, style="generic", type="projection",
mark=NULL, mark.cexmin=0.4, mark.cexmax=1.2, mark.col=1, ...)
```

Arguments

<i>x</i>	Any object of class <i>stpp</i> in spatio-temporal point format.
<i>s.region</i>	Two-column matrix specifying polygonal region containing all data locations. If <i>s.region</i> is missing, the default limits are considered.
<i>t.region</i>	Vector containing the minimum and maximum values of the time interval. If <i>t.region</i> is missing, the default limits are considered.
<i>type</i>	Specify the kind of graphical representation. If <i>type</i> ="projection" (default) the function plot the <i>xy</i> -locations and cumulative distribution of the times. If <i>type</i> ="mark" the function plot the time-mark and space-mark. If <i>type</i> ="scatter" the function plot space-time 3D scatter.
<i>style</i>	Two different classes of graphic styles can be chosen. If <i>style</i> ="generic" (default) the graphics are plot by default function plot in R and if <i>type</i> ="elegant" the graphics are plot based on the R packages <i>ggplot2</i> and <i>plot3D</i> .
<i>mark</i>	Logical. If NULL (default), <i>xy</i> -locations and cumulative distribution of the times are plotted. If TRUE, the time is treated as a quantitative mark attached to each location, and the locations are plotted with the size and/or colour of the plotting symbol determined by the value of the mark.
<i>mark.cexmin</i> , <i>mark.cexmax</i>	Range of the size of the plotting symbol when <i>mark</i> =TRUE.
<i>mark.col</i>	Colour of the plotting symbol when <i>mark</i> =TRUE. If <i>mark.col</i> =0, all locations have the same colour specified by the usual <i>col</i> argument. Otherwise, can be 1 or "black" (default), 2 or "red", 3 or "green", 4 or "blue", in which cases symbols colour is faded, and the darker corresponds to the most recent time.
...	Further arguments to be passed to the functions <i>plot</i> and <i>scatter3D</i> . Typical arguments are <i>pch</i> , <i>theta</i> and <i>phi</i> .

Value

None

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr> and Francisco J. Rodriguez-Cortes.

References

Gabriel E., Rowlingson B., Diggle P. (2013). stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software, 53(2), 1–29.

Stoyan, D., Rodriguez-Cortes, F. J., Mateu, J., and Gille, W. (2017). Mark variograms for spatio-temporal point processes. Spatial Statistics. 20, 125-147.

See Also

[as.3dpoints](#) for creating data in spatio-temporal point format.

 plotK

Plot the estimation of the Space-Time Inhomogeneous K-function

Description

Contour plot or perspective plot or image of the Space-Time Inhomogeneous K-function estimate.

Usage

```
plotK(K,n=15,L=FALSE,type="contour",legend=TRUE,which=NULL,
      main=NULL,...)
```

Arguments

K	Result of the STIKhat function.
n	Number of contour levels desired.
L	Logical indicating whether $K_{ST}(u, v)$ or $L(u, v) = K_{ST}(u, v) - \pi u^2 v$ must be plotted.
type	Specifies the kind of plot: contour by default, but can also be persp or image
legend	Logical indicating whether a legend must be added to the plot.
which	A character specifying the edge correction among the ones used in STIKhat. If a single edge correction method was used in STIKhat, it is not necessary to specify which.
main	Plot title.
...	Additional arguments to persp if persp=TRUE, such as theta and phi.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>

See Also

[contour](#), [persp](#), [image](#) and [STIKhat](#) for an example.

plotPCF

Plot the estimation of the Space-Time Inhomogeneous Pair Correlation function

Description

Contour, image or perspective plot of the Space-Time Inhomogeneous Pair correlation function estimate.

Usage

```
plotPCF(PCF, n=15, type="contour", legend=TRUE, which=NULL,
        main=NULL, ...)
```

Arguments

PCF	Result of the PCFhat function.
n	Number of contour levels desired.
type	Specifies the kind of plot: contour by default, but can also be persp or image
legend	Logical indicating whether a legend must be added to the plot.
which	A character specifying the edge correction among the ones used in PCFhat. If a single edge correction method was used in PCFhat, it is not necessary to specify which.
main	Plot title.
...	Additional arguments to persp if persp=TRUE, such as theta and phi.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>

See Also

[contour](#), [persp](#), [image](#) and [PCFhat](#) for an example.

rinfec *Generate infection point patterns*

Description

Generate one (or several) realisation(s) of the infection process in a region $S \times T$.

Usage

```
rinfec(npoints, s.region, t.region, nsim=1, alpha, beta, gamma,
s.distr="exponential", t.distr="uniform", maxrad, delta, h="step",
g="min", recent=1, lambda=NULL, lmax=NULL, nx=100, ny=100, nt=1000,
t0, inhibition=FALSE, ...)
```

Arguments

npoints	Number of points to simulate.
s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval [0, 1] is considered.
nsim	Number of simulations to generate. Default is 1.
alpha	Numerical value for the latent period.
beta	Numerical value for the maximum infection rate.
gamma	Numerical value for the infection period.
h	Infection rate function which depends on alpha, beta and delta. Must be chosen among "step" and "gaussian".
s.distr	Spatial distribution. Must be chosen among "uniform", "gaussian", "exponential" and "poisson".
t.distr	Temporal distribution. Must be chosen among "uniform" and "exponential".
maxrad	Single value or 2-vector of spatial and temporal maximum radiation respectively. If single value, the same value is used for space and time.
delta	Spatial distance of inhibition/contagion. If missing, the spatial radiation is used.
g	Compute the probability of acceptance of a new point from h and recent. Must be chosen among "min", "max" and "prod".
recent	If "all" consider all previous events. If is an integer, say N , consider only the N most recent events.
lambda	Function or matrix defining the intensity of a Poisson process if s.distr is Poisson.
lmax	Upper bound for the value of lambda.
nx, ny	Define the 2-D grid on which the intensity is evaluated if s.distr is Poisson.
nt	Used to discretize time to compute the infection rate function.

<code>t0</code>	Minimum time used to compute the infection rate function. Default is the minimum of <code>t.region</code> .
<code>inhibition</code>	Logical. If TRUE, an inhibition process is generated. Otherwise, it is a contagious process.
<code>...</code>	Additional parameters if <code>lambda</code> is a function.

Value

A list containing:

<code>xyt</code>	Matrix (or list of matrices if <code>nsim>1</code>) containing the points (x, y, t) of the simulated point pattern. <code>xyt</code> (or any element of the list if <code>nsim>1</code>) is an object of the class <code>stpp</code> .
<code>s.region, t.region</code>	Parameters passed in argument.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>, Peter J Diggle.

See Also

[plot.stpp](#), [animation](#) and [stan](#) for plotting space-time point patterns.

Examples

```
# inhibition; spatial distribution: uniform
inf1 = rinfec(npoints=100, alpha=0.2, beta=0.6, gamma=0.5,
maxrad=c(0.075,0.5), t.region=c(0,50), s.distr="uniform",
t.distr="uniform", h="gaussian", p="min", recent="all", t0=0.02,
inhibition=TRUE)
plot(inf1$xyt, style="elegant")

# contagion; spatial distribution: Poisson with intensity a given matrix
data(fmd)
data(northcumbria)
h = mse2d(as.points(fmd[,1:2]), northcumbria, nsmse=30, range=3000)
h = h$h[which.min(h$mse)]
Ls = kernel2d(as.points(fmd[,1:2]), northcumbria, h, nx=50, ny=50)
inf2 = rinfec(npoints=100, alpha=4, beta=0.6, gamma=20, maxrad=c(12000,20),
s.region=northcumbria, t.region=c(1,2000), s.distr="poisson",
t.distr="uniform", h="step", p="min", recent=1,
lambda=Ls$z, inhibition=FALSE)

image(Ls$x, Ls$y, Ls$z, col=grey((1000:1)/1000)); polygon(northcumbria,lwd=2)
animation(inf2$xyt, add=TRUE, cex=0.7, runtime=15)
```

rinter

*Generate interaction point patterns***Description**

Generate one (or several) realisation(s) of the inhibition or contagious process in a region $S \times T$.

Usage

```
rinter(npoints,s.region,t.region,hs="step",gs="min",thetas=0,
deltas,ht="step",gt="min",thetat=1,deltat,recent="all",nsim=1,
discrete.time=FALSE,replace=FALSE,inhibition=TRUE,...)
```

Arguments

npoints	Number of points to simulate.
s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval $[0, 1]$ is considered.
hs, ht	Function which depends on the distance between points and theta. Can be chosen among "step" and "gaussian" or can refer to a user defined function which only depend on d, theta, and delta (see details). If inhibition=TRUE, h is monotone, increasing, and must tend to 1 when the distance tends to infinity. $0 \leq h(d, \theta) \leq 1$. Otherwise, h is monotone, decreasing, and must tend to 1 when the distance tends to 0.
thetas, thetat	Parameters of hs and ht functions.
deltas, deltat	Spatial and temporal distance of inhibition.
gs, gt	Compute the probability of acceptance of a new point from hs or ht and recent. Must be choosen among "min", "max" and "prod".
recent	If "all" consider all previous events. If is an integer, say N , consider only the N most recent events.
nsim	Number of simulations to generate. Default is 1.
discrete.time	If TRUE, times belong to N , otherwise belong to \mathbf{R}^+ .
replace	Logical. If TRUE allows times repeat.
inhibition	Logical. If TRUE, an inhibition process is generated. Otherwise, it is a contagious process.
...	Additional parameters if hs and ht are defined by the user.

Value

A list containing:

`xyt` Matrix (or list of matrices if `nsim>1`) containing the points (x, y, t) of the simulated point pattern. `xyt` (or any element of the list if `nsim>1`) is an object of the class `stpp`.

`s.region, t.region` Parameters passed in argument.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>, Peter J Diggle.

See Also

[plot.stpp](#), [animation](#) and [stan](#) for plotting space-time point patterns.

Examples

```
# simple inhibition process
inh1 = rinter(npoints=200, thetas=0, deltas=0.05, thetat=0, deltat=0.001,
inhibition=TRUE)
plot(inh1$xyt, style="elegant")
```

```
# inhibition process using hs and ht defined by the user
hs = function(d, theta, delta, mus=0.1)
{
  res=NULL
  a=(1-theta)/mus
  b=theta-a*delta
  for(i in 1:length(d))
  {
    if (d[i]<=delta) res=c(res, theta)
    if (d[i]>(delta+mus)) res=c(res, 1)
    if (d[i]>delta & d[i]<=(delta+mus)) res=c(res, a*d[i]+b)
  }
  return(res)
}
ht = function(d, theta, delta, mut=0.3)
{
  res=NULL
  a=(1-theta)/mut
  b=theta-a*delta
  for(i in 1:length(d))
  {
    if (d[i]<=delta) res=c(res, theta)
    if (d[i]>(delta+mut)) res=c(res, 1)
    if (d[i]>delta & d[i]<=(delta+mut)) res=c(res, a*d[i]+b)
  }
  return(res)
}
```

```

d=seq(0,1,length=100)
plot(d,hs(d,theta=0.2,delta=0.1,mus=0.1),xlab="",ylab="",type="l",
ylim=c(0,1),lwd=2,las=1)
lines(d,ht(d,theta=0.1,delta=0.05,mu=0.3),col=2,lwd=2)
legend("bottomright",col=1:2,lty=1,lwd=2,legend=c(expression(h[s]),
expression(h[t])),bty="n",cex=2)

inh2 = rinter(npoints=100, hs=hs, gs="min", thetas=0.2, deltas=0.1,
ht=ht, gt="min", thetat=0.1, deltat=0.05, inhibition=TRUE)
animation(inh2$xyt, runtime=15, cex=0.8)

# simple contagious process for given spatial and temporal regions
data(northcumbria)
cont1 = rinter(npoints=100, s.region=northcumbria, t.region=c(1,200),
thetas=0, deltas=10000, thetat=0, deltat=10, recent=1, inhibition=FALSE)
plot(cont1$xyt,pch=19,s.region=cont1$s.region,mark=TRUE,mark.col=4)

```

rlgcp

Generate log-Gaussian Cox point patterns

Description

Generate one (or several) realisation(s) of the log-Gaussian cox process in a region $S \times T$.

Usage

```

rlgcp(s.region, t.region, replace=TRUE, npoints=NULL, nsim=1,
nx=100, ny=100, nt=100,separable=TRUE,model="exponential",
param=c(1,1,1,1,1,1), scale=c(1,1),var.grf=1,mean.grf=0,
lmax=NULL,discrete.time=FALSE,exact=FALSE,anisotropy=FALSE,ani.pars=NULL)

```

Arguments

s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval $[0, 1]$ is considered.
npoints	Number of points to simulate. If NULL, the number of points is from a Poisson distribution with mean the double integral of $\lambda(s, t)$ over s.region and t.region.
nsim	number of simulations to generate. Default is 1.
separable	Logical. If TRUE, the covariance function of the Gaussian random field is separable.

model	Vector of length 1 or 2 specifying the model(s) of covariance of the Gaussian random field. If separable=TRUE and model is of length 2, then the elements of model define the spatial and temporal covariances respectively. If separable=TRUE and model is of length 1, then the spatial and temporal covariances belongs to the same class of covariances, among "matern", "exponential", "stable", "cauchy" and "wave" (see Details). If separable=FALSE, model must be of length 1 and is either "gneiting" or "cesare" (see Details).
param	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$. Vector of parameters of the covariance function (see Details).
scale	Vector of length 2 defining the spatial and temporal scale.
var.grf	Variance of the Gaussian random field.
mean.grf	Mean of the Gaussian random field.
replace	Logical allowing times repeat.
nx, ny, nt	Define the size of the 3-D grid on which the intensity is evaluated.
lmax	Upper bound for the value of $\lambda(x, y, t)$.
discrete.time	If TRUE, times belong to N , otherwise belong to R^+ .
exact	logical allowing exact simulation of Gaussian random fields (see manual for details).
anisotropy	If TRUE, simulate an anisotropic point pattern. Currently only implemented for separable covariance functions.
ani.pars	Vector of length 2, the anisotropy angle and the anisotropy ratio, respectively (see details).

Details

We implemented stationary, isotropic spatio-temporal covariance functions.

Separable covariance functions

$$c(h, t) = c_s(\|h\|) c_t(|t|), h \in S, t \in T$$

The purely spatial and purely temporal covariance functions can be:

- Exponential: $c(r) = \exp(-r)$,
- Stable: $c(r) = \exp(-r^\alpha)$, $\alpha \in [0, 2]$,
- Cauchy: $c(r) = (1 + r^2)^{-\alpha}$, $\alpha > 0$,
- Wave: $c(r) = \frac{\sin(r)}{r}$ if $r > 0$, $c(0) = 1$,
- Matern: $c(r) = \frac{(\alpha r)^\nu}{2^{\nu-1}\Gamma(\nu)} \mathcal{K}_\nu(\alpha r)$, $\nu > 0$ and $\alpha > 0$.

\mathcal{K}_ν is the modified Bessel function of second kind:

$$\mathcal{K}_\nu(x) = \frac{\pi}{2} \frac{I_{-\nu}(x) - I_\nu(x)}{\sin(\pi\nu)},$$

with $I_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{1}{k!\Gamma(\nu+k+1)} \left(\frac{x}{2}\right)^{2k}$.

The parameters α_1 and α_2 correspond to the parameters of the spatial and temporal covariance respectively. For the Matern model, the parameters α_1 , α_3 and α_2 , α_4 correspond to the parameters ν , α of the spatial and temporal covariance.

Non-separable covariance functions

The spatio-temporal covariance function can be:

- Gneiting: $c(h, t) = \psi(t^2/\beta_2)^{-\alpha_6} \phi\left(\frac{h^2/\beta_1}{\psi(t^2/\beta_2)}\right)$, $\beta_1, \beta_2 > 0$ are spatial and temporal scales respectively,
 - If $\alpha_2 = 1$, $\phi(r)$ is the Stable model.
 - if $\alpha_2 = 2$, $\phi(r)$ is the Cauchy model.
 - If $\alpha_2 = 3$, $\phi(r)$ is the Matern model.
 - If $\alpha_5 = 1$, $\psi(r) = (r^{\alpha_3} + 1)^{\alpha_4}$,
 - If $\alpha_5 = 2$, $\psi(r) = (\alpha_4^{-1}r^{\alpha_3} + 1)/(r^{\alpha_3} + 1)$,
 - If $\alpha_5 = 3$, $\psi(r) = \log(r^{\alpha_3} + \alpha_4)/\log \alpha_4$,

The parameter α_1 is the respective parameter for the model of $\phi(\cdot)$, $\alpha_3 \in (0, 1]$, $\alpha_4 \in (0, 1]$ and $\alpha_6 \geq 2$.

- cesare: $c(h, t) = (1 + (h/\beta_1)^{\alpha_1} + (t/\beta_2)^{\alpha_2})^{-\alpha_3}$, $\beta_1, \beta_2 > 0$, $\alpha_1, \alpha_2 \in [1, 2]$ and $\alpha_3 \geq 3/2$.

We also implemented anisotropic Log-Gaussian Cox processes. We considered geometric spatial anisotropy (see Moller and Toftaker, 2014). In this case the covariance function is elliptical and anisotropy is characterized by two parameters: the anisotropy angle $0 \leq \theta < \pi$ and the anisotropy ratio $0 < \delta \leq 1$ of the minor axis $2\omega\delta$ and the major axis 2ω .

$$C(h, t) = C_0 \left(\sqrt{h\Sigma^{-1}h'}, t \right), h \in R^2.$$

Value

A list containing:

xyt	Matrix (or list of matrices if <code>nsim>1</code>) containing the points (x, y, t) of the simulated point pattern. xyt (or any element of the list if <code>nsim>1</code>) is an object of the class <code>stpp</code> .
s.region, t.region	parameters passed in argument.
Lambda	$nx \times ny \times nt$ array (or list of array if <code>nsim>1</code>) of the intensity.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>, Peter J Diggle.

References

- Chan, G. and Wood A. (1997). An algorithm for simulating stationary Gaussian random fields. *Applied Statistics*, Algorithm Section, 46, 171–181.
- Chan, G. and Wood A. (1999). Simulation of stationary Gaussian vector fields. *Statistics and Computing*, 9, 265–268.
- Gneiting T. (2002). Nonseparable, stationary covariance functions for space-time data. *Journal of the American Statistical Association*, 97, 590–600.
- Moller J. and Toftaker H. (2014). Geometric anisotropic spatial point pattern analysis and Cox processes. *Scandinavian Journal of Statistics*, 41, 414–435.

See Also

[plot.stpp](#), [animation](#) and [stan](#) for plotting space-time point patterns.

Examples

```
# non separable covariance function:
lgcp1 <- rlgcp(npoints=200, nx=50, ny=50, nt=50, separable=FALSE,
model="gneiting", param=c(1,1,1,1,1,2), var.grf=1, mean.grf=0)
N <- lgcp1$Lambda[, ,1];for(j in 2:(dim(lgcp1$Lambda)[3])){N <-
N+lgcp1$Lambda[, ,j]}
image(N,col=grey((1000:1)/1000));box()
animation(lgcp1$xyt, cex=0.8, runtime=10, add=TRUE, prevalent="orange")

# separable covariance function:
lgcp2 <- rlgcp(npoints=200, nx=50, ny=50, nt=50, separable=TRUE,
model="exponential", param=c(1,1,1,1,1,2), var.grf=2, mean.grf=-0.5*2)
N <- lgcp2$Lambda[, ,1];for(j in 2:(dim(lgcp2$Lambda)[3])){N <-
N+lgcp2$Lambda[, ,j]}
image(N,col=grey((1000:1)/1000));box()
animation(lgcp2$xyt, cex=0.8, pch=20, runtime=10, add=TRUE,
prevalent="orange")

# anisotropic

sigma2=0.5
simlgcp <- rlgcp(npoints=500,nx=250, ny=200, nt=50,separable=TRUE,
s.region=matrix(c(0,2,2,0,0,0,0.5,0.5),byrow=FALSE,ncol=2), model="exponential",
param=c(1,1,1,1,1,2), var.grf=sigma2, mean.grf=-0.5*sigma2,anisotropy = TRUE,
ani.pars = c(pi/4,0.1))

N <- simlgcp$Lambda[, ,1];for(j in 2:dim(simlgcp$Lambda)[3]){N <- N+simlgcp$Lambda[, ,j]}
image(x=simlgcp$grid[[1]]$x,y=simlgcp$grid[[1]]$y,z=N,col=grey((1000:1)/1000));box()
points(simlgcp$xyt[,1:2],pch=19,cex=0.25,col=2)
```

 rpcp

Generate Poisson cluster point patterns

Description

Generate one (or several) realisation(s) of the Poisson cluster process in a region $S \times T$.

Usage

```
rpcp(s.region, t.region, nparents=NULL, npoints=NULL, lambda=NULL,
mc=NULL, nsim=1, cluster="uniform", dispersion, infectious=TRUE,
edge = "larger.region", larger.region=larger.region, tronc=1,...)
```

Arguments

<code>s.region</code>	Two-column matrix specifying polygonal region containing all data locations. If <code>s.region</code> is missing, the unit square is considered.
<code>t.region</code>	Vector containing the minimum and maximum values of the time interval. If <code>t.region</code> is missing, the interval $[0, 1]$ is considered.
<code>nparents</code>	Number of parents. If NULL, <code>nparents</code> is from a Poisson distribution with intensity <code>lambda</code> .
<code>npoints</code>	Number of points to simulate. If NULL (default), the number of points is from a Poisson distribution with mean the double integral of the intensity over <code>s.region</code> and <code>t.region</code> .
<code>lambda</code>	Intensity of the parent process. Can be either a numeric value, a function, or a 3d-array (see <code>rpp</code>). If NULL, it is constant and equal to <code>nparents</code> / volume of the domain.
<code>mc</code>	Average number of children per parent. It is used when <code>npoints</code> is NULL.
<code>nsim</code>	Number of simulations to generate.
<code>cluster</code>	Distribution of children: “uniform”, “normal” and “exponential” are currently implemented. Either a single value if the distribution in space and time is the same, or a vector of length 2, giving first the spatial distribution of children and then the temporal distribution.
<code>dispersion</code>	Scale parameter. It equals twice the standard deviation of location of children relative to their parent for a normal distribution of children; the mean for an exponential distribution and half range for an uniform distribution.
<code>infectious</code>	If TRUE, offspring’s times are always greater than parent’s time.
<code>edge</code>	Specify the edge correction to use “larger.region” or “without”.
<code>larger.region</code>	By default, the larger spatial region is the convex hull of <code>s.region</code> enlarged by the spatial related value of <code>dispersion</code> and the larger time interval is <code>t.region</code> enlarged by the temporal related value of <code>dispersion</code> . One can over-ride default using the 2-vector parameter <code>larger.region</code> .
<code>tronc</code>	Parameter of the truncated exponential distribution for the distribution of children.
...	Additional parameters of the intensity of the parent process.

Value

A list containing:

<code>xyt</code>	Matrix (or list of matrices if <code>nsim</code> >1) containing the points (x, y, t) of the simulated point pattern. <code>xyt</code> (or any element of the list if <code>nsim</code> >1) is an object of the class <code>stpp</code> .
<code>s.region, t.region</code>	Parameters passed in argument.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>, Peter J Diggle.

See Also

[plot.stpp](#), [animation](#) and [stan](#) for plotting space-time point patterns.

Examples

```
# homogeneous Poisson distribution of parents

data(northcumbria)
pcp1 <- rpcp(nparents=50, npoints=500, s.region=northcumbria,
t.region=c(1,365), cluster=c("normal","exponential"),
maxrad=c(5000,5))

animation(pcp1$xyt, s.region=pcp1$s.region, t.region=pcp1$t.region,
runtime=5)

# inhomogeneous Poisson distribution of parents

lbda <- function(x,y,t,a){a*exp(-4*y) * exp(-2*t)}
pcp2 <- rpcp(nparents=50, npoints=500, cluster="normal", lambda=lbda,
a=4000/((1-exp(-4))*(1-exp(-2))))
plot(pcp2$xyt, style="elegant")
```

rpp

Generate Poisson point patterns

Description

Generate one (or several) realisation(s) of the (homogeneous or inhomogeneous) Poisson process in a region $S \times T$.

Usage

```
rpp(lambda, s.region, t.region, npoints=NULL, nsim=1, replace=TRUE,
discrete.time=FALSE, nx=100, ny=100, nt=100, lmax=NULL, ...)
```

Arguments

lambda	Spatio-temporal intensity of the Poisson process. If lambda is a single positive number, the function generates realisations of a homogeneous Poisson process, whilst if lambda is a function of the form $\lambda(x, y, t, \dots)$ or a 3D-array it generates realisations of an inhomogeneous Poisson process.
s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval $[0, 1]$ is considered.
replace	Logical allowing times repeat (should only be used when discrete.time=TRUE).

<code>npoints</code>	Number of points to simulate. If NULL, the number of points is from a Poisson distribution with mean the double integral of <code>lambda</code> over <code>s.region</code> and <code>t.region</code> .
<code>discrete.time</code>	If TRUE, times belong to \mathbf{N} , otherwise belong to \mathbf{R}^+ .
<code>nsim</code>	Number of simulations to generate. Default is 1.
<code>nx,ny,nt</code>	Define the size of the 3-D grid on which the intensity is evaluated.
<code>lmax</code>	Upper bound for the value of $\lambda(x, y, t)$, if <code>lambda</code> is a function.
<code>...</code>	Additional parameters if <code>lambda</code> is a function.

Value

A list containing:

<code>xyt</code>	Matrix (or list of matrices if <code>nsim>1</code>) containing the points (x, y, t) of the simulated point pattern. <code>xyt</code> (or any element of the list if <code>nsim>1</code>) is an object of the class <code>stpp</code> .
<code>Lambda</code>	$nx \times ny \times nt$ array of the intensity surface at each time.
<code>s.region, t.region, lambda</code>	parameters passed in argument.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr> and Peter J Diggle.

See Also

[plot.stpp](#), [animation](#) and [stan](#) for plotting space-time point patterns.

Examples

```
# Homogeneous Poisson process
# -----
hpp1 <- rpp(lambda=200,replace=FALSE)

stan(hpp1$xyt)

# fixed number of points, discrete time, with time repeat.
data(northcumbria)
hpp2 <- rpp(npoints=500, s.region=northcumbria, t.region=c(1,1000),
discrete.time=TRUE)
plot(hpp2$xyt, style="elegant")

polymap(northcumbria)
animation(hpp2$xyt, s.region=hpp2$s.region, t.region=hpp2$t.region,
runtime=10, add=TRUE)

# Inhomogeneous Poisson process
```

```

# -----

# intensity defined by a function
lbda1 = function(x,y,t,a){a*exp(-4*y) * exp(-2*t)}
ipp1 = rpp(lambda=lbda1, npoints=400, a=3200/((1-exp(-4))*(1-exp(-2))))
stan(ipp1$xyt)

# intensity defined by a matrix
data(fmd)
data(northcumbria)
h = mse2d(as.points(fmd[,1:2]), northcumbria, nsmse=30, range=3000)
h = h$h[which.min(h$mse)]
Ls = kernel2d(as.points(fmd[,1:2]), northcumbria, h, nx=100, ny=100)
Lt = dim(fmd)[1]*density(fmd[,3], n=200)$y
Lst=array(0,dim=c(100,100,200))
for(k in 1:200) Lst[, ,k] <- Ls*z*Lt[k]/dim(fmd)[1]
ipp2 = rpp(lambda=Lst, s.region=northcumbria, t.region=c(1,200),
discrete.time=TRUE)

par(mfrow=c(1,1))
image(Ls$x, Ls$y, Ls$z, col=grey((1000:1)/1000)); polygon(northcumbria)
animation(ipp2$xyt, add=TRUE, cex=0.5, runtime=15)

```

sim.stpp

Generate spatio-temporal point patterns

Description

Generate one (or several) realisation(s) of a spatio-temporal point process in a region $S \times T$.

Usage

```
sim.stpp(class="poisson", s.region, t.region, npoints=NULL,
nsim=1, ...)
```

Arguments

<code>class</code>	Must be chosen among "poisson", "cluster", "cox", "infectious" and "inhibition".
<code>s.region</code>	Two-column matrix specifying polygonal region containing all data locations. If <code>s.region</code> is missing, the unit square is considered.
<code>t.region</code>	Vector containing the minimum and maximum values of the time interval. If <code>t.region</code> is missing, the interval $[0, 1]$ is considered.
<code>npoints</code>	Number of points to simulate.
<code>nsim</code>	Number of simulations to generate. Default is 1.
<code>...</code>	Additional parameters related to the <code>class</code> parameter. See rpp for the Poisson process; rpcp for the Poisson cluster process; rlgcp for the Log-Gaussian Cox process; rinter for the interaction (inhibition or contagious) process and rinfec for the infectious process.

Value

A list containing:

`xyt` Matrix (or list of matrices if `nsim>1`) containing the points (x, y, t) of the simulated point pattern. `xyt` (or any element of the list if `nsim>1`) is an object of the class `stpp`.

`s.region`, `t.region`
Parameters passed in argument.

Author(s)

Edith Gabriel <edith.gabriel@inrae.fr>

See Also

[rpp](#), [rinfec](#), [rinter](#), [rpcp](#) and [rlgcp](#) for the simulation of Poisson, infectious, interaction, Poisson cluster and log-gaussian Cox processes respectively; and [plot.stpp](#), [animation](#) and [stan](#) for plotting space-time point patterns.

`stan` *(3D) space-time data animation*

Description

Displays (x, y, t) point data and enables dynamic highlighting of time slices.

Usage

```
stan(xyt, tlim=range(xyt[,3]), na.rm=TRUE), twid=diff(tlim)/20,
persist=FALSE, states, bgpoly, bgframe=TRUE, bgimage,
bgcol=gray(seq(0,1,len=12)), axes=TRUE)
```

Arguments

`xyt` A 3-column matrix of x, y, t coordinates

`tlim` A two-element vector of upper and lower time limits

`twid` The initial time window width

`persist` Whether to display points before time window

`states` How to display points - see Details

`bgpoly` A polygon to draw on the background plane

`bgframe` Whether to extend the `bgpoly` to the front plane

`bgimage` An list with x, y vectors and z matrix to display on the background plane

`bgcol` A colour palette vector with which to draw the `bgimage`

`axes` Logical value indicating whether labels should be added.

Details

This function requires the `rpanel` and `rgl` packages. It uses `rpanel` for the sliders to control the graphics, and `rgl` for its ability to do flicker-free graphics.

The sliders set the position and width of the temporal highlight window. For 'time' slider set to time T and 'width' slider set to S , highlighted points are those with time coordinate t such that $T - S < t < T$.

How points are shown is configured with the `states` parameter. This is a list of length 3 specifying how points before the time window, inside the time window, and after the time window are displayed. Each element is a list of parameters as would be passed to `material3d()` together with a radius element. Points are drawn as spheres with the corresponding material and radius as a fraction of the spatial span of the data.

By default the third state is invisible, and the first two states are different. By calling with the default for `states` and `persist=TRUE`, then the first state is set to the same as the second state. This has the effect of showing all points at time $< T$ with the same sphere type.

If the user specifies the `states` parameter, then `persist` is ignored. The user can emulate the `persist` behaviour by specifying a `states` list with identical parameters for states 1 and 2.

Note that each state element should specify all `material3d` parameters used in any of the state elements. This is to make sure the parameters are reset for each of the sets of points.

The background polygon must be a simple 2-column vector of x and y coordinates. When used with `bgframe=TRUE`, the polygon is also drawn on the front plane, and the convex hull points are connected front to back in order to visualise the space-time prism that the data are contained in.

A raster image can be displayed on the back plane by setting the `bgimage` parameter. This must be a list with x , y and z components as needed by the image function. Note that x and y define the center of cells and so must be the same length as the dimensions of z - the image function can accept x and y values that are one longer than the dimensions of z to define the edges, but `bgimage` does not allow that.

Value

A list of the slider parameters when the dialog is quitted.

Author(s)

Barry Rowlingson <b.rowlingson@lancaster.ac.uk>, Edith Gabriel

stdcpp

Generate double-cluster point pattern

Description

Generate a realisation of the double-cluster process in a region $S \times T$.

Usage

```
stdcpp(lambp, a, b, c, mu, s.region, t.region)
```

Arguments

s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval [0, 1] is considered.
lambp	Intensity of the parent process. Can be either a numeric value, a function, or a 3d-array (see rpp).
a	Length of the semi-axes x of ellipsoid.
b	Length of the semi-axes y of ellipsoid.
c	Length of the semi-axes z of ellipsoid.
mu	Average number of daughter per parent. (a single positive number).

Details

We consider the straightforward extension of the classical Matern cluster process on the R^3 case (with ellipsoid or balls) by considering the z -coordinates as times.

Consider a Poisson point process in the plane with intensity λ_p as cluster centres for all times 'parent', as well as a ellipsoid (or ball) where the semi-axes are of lengths a , b and c , around of each Poisson point under a random general rotation. The scatter uniformly in all ellipsoid (or ball) of all points which are of the form (x, y, z) , the number of points in each cluster being random with a Poisson (μ) distribution. The resulting point pattern is a spatio-temporal cluster point process with $t = z$. This point process has intensity $\lambda_p \times \mu$.

Value

The simulated spatio-temporal point pattern.

Author(s)

Francisco J. Rodriguez Cortes <frrodriguez@unal.edu.co>

References

- Baddeley, A., Rubak, E., Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, Boca Raton.
- Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). *Stochastic Geometry and its Applications*. John Wiley & Sons.
- Gabriel, E., Rowlingson, B., Diggle P J. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. *Journal of Statistical Software* 53, 1-29.
- Illian, J B., Penttinen, A., Stoyan, H. and Stoyan, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons, London.
- Stoyan, D., Rodriguez-Cortes, F. J., Mateu, J., and Gille, W. (2017). Mark variograms for spatio-temporal point processes. *Spatial Statistics*. 20, 125-147.

Examples

```

# Ellipsoid
Xe <- stdcpp(lambp=20,a=0.5,b=0.09,c=0.07,mu=100)
plot(Xe$xyt)

# Spatio-temporal 3D scatter plot
par(mfrow=c(1,1))
plot(Xe$xyt,type="scatter")

# Balls
Xb <- stdcpp(lambp=20,a=0.07,b=0.07,c=0.07,mu=100)
plot(Xb$xyt)

# Spatio-temporal 3D scatter plot
par(mfrow=c(1,1))
plot(Xb$xyt,type="mark",style="elegant")

# Northcumbria
data(northcumbria)
Northcumbria <- northcumbria/1000
X <- stdcpp(lambp=0.00004,a=10,b=10,c=10,mu=120,
s.region=Northcumbria,t.region=c(0,200))
plot(X$xyt,s.region=Northcumbria, cex=0.5)

# Spatio-temporal 3D scatter plot
par(mfrow=c(1,1))
plot(X$xyt,type="scatter",theta=45,phi=30,cex=0.1,
ticktype="detailed",col="black",style="elegant")

```

sthpcpp

*Spatio-temporal hot-spots cluster point process model***Description**

Generate a realisation of the hot-spots cluster process in a region $S \times T$.

Usage

```
sthpcpp(lambp, r, mu, s.region, t.region)
```

Arguments

s.region	Two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered.
t.region	Vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval $[0, 1]$ is considered.
lambp	Intensity of the Poisson process of cluster centres. A single positive number, a function, or a pixel image.

r	Radius parameter of the clusters.
mu	Average number of daughter per parent (a single positive number) or reference intensity for the cluster points (a function or a pixel image).

Details

This function generates a realisation of spatio-temporal cluster process, which can be considered as generalisation of the classical Matern cluster process, inside the spatio-temporal window.

Consider a Poisson point process in the plane with intensity λ_p as cluster centres for all times 'parent', as well as a infinite cylinder of radius R around of each Poisson point, orthogonal to the plane. The scatter uniformly in all cylinders of all points which are of the form (x, y, z) , the number of points in each cluster being random with a Poisson (μ) distribution. The resulting point pattern is a spatio-temporal cluster point process with $t = z$. This point process has intensity $\lambda_p \times \mu$.

Value

The simulated spatio-temporal point pattern.

Author(s)

Francisco J. Rodriguez Cortes <frodriguez@unal.edu.co>

References

- Baddeley, A., Rubak, E., Turner, R. (2015). Spatial Point Patterns: Methodology and Applications with R. CRC Press, Boca Raton.
- Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). Stochastic Geometry and its Applications. John Wiley & Sons.
- Gabriel, E., Rowlingson, B., Diggle P J. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software 53, 1-29.
- Illian, J B., Penttinen, A., Stoyan, H. and Stoyan, D. (2008). Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley and Sons, London.
- Stoyan, D., Rodriguez-Cortes, F. J., Mateu, J., and Gille, W. (2017). Mark variograms for spatio-temporal point processes. Spatial Statistics. 20, 125-147.

Examples

```
# First example
X <- sthpcpp(lambp=20, r=0.05, mu=100)
plot(X$xyt)

# Spatio-temporal 3D scatter plot
par(mfrow=c(1,1))
plot(X$xyt,type="scatter")

## Spatio-temporal hot-spots cluster point process model
data(northcumbria)
Northcumbria <- northcumbria/1000
Xo <- sthpcpp(lambp=0.0035, r=5, mu=200,
```

```
s.region=Northcumbria, t.region=c(28,198))
plot(Xo$xyt,s.region=Northcumbria)

# Spatio-temporal 3D scatter plot
par(mfrow=c(1,1))
plot(Xo$xyt,type="scatter",style="elegant",theta=45,phi=30,cex=0.05,
ticktype="detailed",col="black")
```

STIKhat

*Estimation of the Space-Time Inhomogeneous K-function***Description**

Compute an estimate of the Space-Time Inhomogeneous K-function.

Usage

```
STIKhat(xyt, s.region, t.region, dist, times, lambda,
correction="isotropic", infectious=FALSE)
```

Arguments

<code>xyt</code>	Coordinates and times (x, y, t) of the point pattern.
<code>s.region</code>	Two-column matrix specifying polygonal region containing all data locations. If <code>s.region</code> is missing, the bounding box of <code>xyt[,1:2]</code> is considered.
<code>t.region</code>	Vector containing the minimum and maximum values of the time interval. If <code>t.region</code> is missing, the range of <code>xyt[,3]</code> is considered.
<code>dist</code>	Vector of distances u at which $K(u, v)$ is computed. If missing, the maximum of <code>dist</code> is given by $\min(S_x, S_y)/4$, where S_x and S_y represent the maximum width and height of the bounding box of <code>s.region</code> .
<code>times</code>	Vector of times v at which $K(u, v)$ is computed. If missing, the maximum of <code>times</code> is given by $(T_{\max} - T_{\min})/4$, where T_{\min} and T_{\max} are the minimum and maximum of the time interval T .
<code>lambda</code>	Vector of values of the space-time intensity function evaluated at the points (x, y, t) in $S \times T$. If <code>lambda</code> is missing, the estimate of the space-time K-function is computed as for the homogeneous case (Diggle et al., 1995), i.e. considering $n/ S \times T $ as an estimate of the space-time intensity.
<code>correction</code>	A character vector specifying the edge correction(s) to be applied among "isotropic", "border", "modified.border", "translate" and "none" (see Details). The default is "isotropic".
<code>infectious</code>	Logical value. If TRUE, only future events are considered and the isotropic edge correction method is used. See Details.

Details

Gabriel (2014) proposes the following unbiased estimator for the STIK-function, based on data giving the locations of events $x_i : i = 1, \dots, n$ on a spatio-temporal region $S \times T$, where S is an arbitrary polygon and T is a time interval:

$$\hat{K}(u, v) = \sum_{i=1}^n \sum_{j \neq i} \frac{1}{w_{ij}} \frac{1}{\lambda(x_i)\lambda(x_j)} \mathbf{1}_{\{\|s_i - s_j\| \leq u ; |t_i - t_j| \leq v\}},$$

where $\lambda(x_i)$ is the intensity at $x_i = (s_i, t_i)$ and w_{ij} is an edge correction factor to deal with spatial-temporal edge effects. The edge correction methods implemented are:

isotropic: $w_{ij} = |S \times T| w_{ij}^{(t)} w_{ij}^{(s)}$, where the temporal edge correction factor $w_{ij}^{(t)} = 1$ if both ends of the interval of length $2|t_i - t_j|$ centred at t_i lie within T and $w_{ij}^{(t)} = 1/2$ otherwise and $w_{ij}^{(s)}$ is the proportion of the circumference of a circle centred at the location s_i with radius $\|s_i - s_j\|$ lying in S (also called Ripley's edge correction factor).

border: $w_{ij} = \frac{\sum_{j=1}^n \mathbf{1}_{\{d(s_j, S) > u ; d(t_j, T) > v\}} / \lambda(x_j)}{\mathbf{1}_{\{d(s_i, S) > u ; d(t_i, T) > v\}}}$, where $d(s_i, S)$ denotes the distance between s_i and the boundary of S and $d(t_i, T)$ the distance between t_i and the boundary of T .

modified.border: $w_{ij} = \frac{|S_{\ominus u}| \times |T_{\ominus v}|}{\mathbf{1}_{\{d(s_i, S) > u ; d(t_i, T) > v\}}}$, where $S_{\ominus u}$ and $T_{\ominus v}$ are the eroded spatial and temporal region respectively, obtained by trimming off a margin of width u and v from the border of the original region.

translate: $w_{ij} = |S \cap S_{s_i - s_j}| \times |T \cap T_{t_i - t_j}|$, where $S_{s_i - s_j}$ and $T_{t_i - t_j}$ are the translated spatial and temporal regions.

none: No edge correction is performed and $w_{ij} = |S \times T|$.

If parameter infectious = TRUE, only future events are considered and the estimator is, using an isotropic edge correction factor (Gabriel and Diggle, 2009):

$$\hat{K}(u, v) = \frac{1}{|S \times T|} \frac{n}{n_v} \sum_{i=1}^{n_v} \sum_{j=1; j > i}^{n_v} \frac{1}{w_{ij}} \frac{1}{\lambda(x_i)\lambda(x_j)} \mathbf{1}_{\{u_{ij} \leq u\}} \mathbf{1}_{\{t_j - t_i \leq v\}}.$$

In this equation, the points $x_i = (s_i, t_i)$ are ordered so that $t_i < t_{i+1}$, with ties due to round-off error broken by randomly unrounding if necessary. To deal with temporal edge-effects, for each v , n_v denotes the number of events for which $t_i \leq T_1 - v$, with $T = [T_0, T_1]$. To deal with spatial edge-effects, we use Ripley's method.

If lambda is missing in argument, STIKhat computes an estimate of the space-time (homogeneous) K-function:

$$\hat{K}(u, v) = \frac{|S \times T|}{n_v(n-1)} \sum_{i=1}^{n_v} \sum_{j=1; j > i}^{n_v} \frac{1}{w_{ij}} \mathbf{1}_{\{u_{ij} \leq u\}} \mathbf{1}_{\{t_j - t_i \leq v\}}$$

Value

A list containing:

Khat ndist x ntimes matrix containing values of $\hat{K}_{ST}(u, v)$.
 Ktheo ndist x ntimes matrix containing theoretical values for a Poisson process; $\pi u^2 v$ for K and $2\pi u^2 v$ for K^* .


```

v <- seq(0,15,by=1)
stik1 <- STIKhat(xyt=FMD, s.region=northcumbria/1000,t.region=c(1,200),
lambda=mhat*mut/dim(FMD)[1], dist=u, times=v, infectious=TRUE)

# plotting the estimation
plotK(stik1)
plotK(stik1,type="persp",theta=-65,phi=35)

# Second example

xyt=rpp(lambda=200)
stik2=STIKhat(xyt$xyt,dist=seq(0,0.16,by=0.02),
times=seq(0,0.16,by=0.02),correction=c("border","translate"))
plotK(stik2,type="contour",legend=TRUE,which="translate")

```

stpp

Space-Time Point Pattern simulation, visualisation and analysis

Description

This package provides models of spatio-temporal point processes in a region $S \times T$ and statistical tools for analysing global and local second-order properties of such processes. It also includes static and dynamic (2D and 3D) plots. `stpp` is the first dedicated unified computational environment in the area of spatio-temporal point processes.

The `stpp` package depends upon some other packages:

`splancs`: spatial and space-time point pattern analysis

`rgl`: interactive 3D plotting of densities and surfaces

`rpanel`: simple interactive controls for R using `tcltk` package

`KernSmooth`: functions for kernel smoothing for Wand & Jones (1995)

`plot3D`: Tools for plotting 3-D and 2-D data

`ggplot2`: Create Elegant Data Visualisations Using the Grammar of Graphics

Details

`stpp` is a package for simulating, analysing and visualising patterns of points in space and time.

Following is a summary of the main functions and the dataset in the `stpp` package.

To visualise a spatio-temporal point pattern

- [animation](#): space-time data animation.
- [as.3dpoints](#): create data in spatio-temporal point format.
- [plot.stpp](#): plot spatio-temporal point object. Either a two-panels plot showing spatial locations and cumulative times, or a one-panel plot showing spatial locations with times treated as a quantitative mark attached to each location.
- [stan](#): 3D space-time animation.

To simulate spatio-temporal point patterns

- `rinfec`: simulate an infection point process,
- `rinter`: simulate an interaction (inhibition or contagious) point process,
- `rlgcp`: simulate a log-Gaussian Cox point process,
- `rpcp`: simulate a Poisson cluster point process,
- `rpp`: simulate a Poisson point process,
- `stdcpp`: simulate a double-cluster point process,
- `sthpcpp`: simulate a hot-spot point process.

To analyse spatio-temporal point patterns

- `PCFhat`: space-time inhomogeneous pair correlation function,
- `STIKhat`: space-time inhomogeneous K-function,
- `ASTIKhat`: Anisotropic space-time inhomogeneous K-function,
- `LISTAhat`: space-time inhomogeneous pair correlation LISTA functions.
- `KLISTAhat`: space-time inhomogeneous K LISTA functions.
- `gsp`: spatial mark variogram function.
- `gte`: Temporal mark variogram function.

Dataset

`fmd`: 2001 food-and-mouth epidemic in north Cumbria (UK).

Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter J. Diggle, Barry Rowlingson and Francisco J. Rodriguez-Cortes

References

- Baddeley, A., Rubak, E., Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, Boca Raton.
- Chan, G. and Wood A. (1997). An algorithm for simulating stationary Gaussian random fields. *Applied Statistics, Algorithm Section*, 46, 171–181.
- Chan, G. and Wood A. (1999). Simulation of stationary Gaussian vector fields. *Statistics and Computing*, 9, 265–268.
- Diggle P. , Chedwynd A., Haggkvist R. and Morris S. (1995). Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, 4, 124–136.
- Diggle, P.J., 2013. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC Press, Boca Raton.
- Gabriel E. (2014). Estimating second-order characteristics of inhomogeneous spatio-temporal point processes: influence of edge correction methods and intensity estimates. *Methodology and computing in Applied Probability*, 16(1).

- Gabriel E., Diggle P. (2009). Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica*, 63, 43–51.
- Gabriel E., Rowlingson B., Diggle P. (2013). stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. *Journal of Statistical Software*, 53(2), 1–29.
- Gneiting T. (2002). Nonseparable, stationary covariance functions for space-time data. *Journal of the American Statistical Association*, 97, 590–600.
- Gonzalez, J. A., Rodriguez-Cortes, F. J., Cronie, O. and Mateu, J. (2016). Spatio-temporal point process statistics: a review. *Spatial Statistics*, 18, 505–544.
- Siino, M., Rodriguez-Cortes, F. J., Mateu, J. and Adelfio, G. (2017). Testing for local structure in spatio-temporal point pattern data. *Environmetrics*. DOI: 10.1002/env.2463.
- Stoyan, D., Rodriguez-Cortes, F. J., Mateu, J., and Gille, W. (2017). Mark variograms for spatio-temporal point processes. *Spatial Statistics*. 20, 125-147.

Index

*Topic **datasets**

fmd, [5](#)

northcumbria, [13](#)

*Topic **hplot**

stan, [32](#)

sthpcpp, [35](#), [41](#)

STIKhat, [4](#), [19](#), [37](#), [41](#)

stpp, [40](#)

stpp-package (stpp), [40](#)

animation, [2](#), [21](#), [23](#), [27](#), [29](#), [30](#), [32](#), [40](#)

as.3dpoints, [3](#), [18](#), [40](#)

ASTIKhat, [4](#), [41](#)

contour, [19](#)

fmd, [5](#), [13](#), [41](#)

gsp, [6](#), [41](#)

gte, [7](#), [41](#)

image, [19](#)

is.3dpoints, [9](#)

KLISTAhat, [10](#), [41](#)

LISTAhat, [11](#), [41](#)

northcumbria, [5](#), [13](#)

PCFhat, [10](#), [12](#), [14](#), [19](#), [41](#)

persp, [19](#)

plot (plot.stpp), [17](#)

plot.stpp, [17](#), [21](#), [23](#), [27](#), [29](#), [30](#), [32](#), [40](#)

plotK, [18](#)

plotPCF, [19](#)

rinfec, [20](#), [31](#), [32](#), [41](#)

rinter, [22](#), [31](#), [32](#), [41](#)

rlgcp, [24](#), [31](#), [32](#), [41](#)

rpcp, [27](#), [31](#), [32](#), [41](#)

rpp, [28](#), [29](#), [31](#), [32](#), [34](#), [41](#)

sim.stpp, [31](#)

stan, [21](#), [23](#), [27](#), [29](#), [30](#), [32](#), [32](#), [40](#)

stdcpp, [33](#), [41](#)