Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

stpp: An R Package for Plotting, Simulating and Analysing Spatio-Temporal Point Patterns

Edith Gabriel
Université d'Avignon
et des Pays de Vaucluse

Barry Rowlingson
Lancaster University

Peter J Diggle
Lancaster University
University of Liverpool

Abstract

stpp is an R package for analysing, simulating and displaying space-time point patterns. It covers many of the models encountered in applications of point process methods to the study of spatio-temporal phenomena. The package also includes estimators of the space-time inhomogeneous K-function and pair correlation function. **stpp** is the first dedicated unified computational environment in the area of spatio-temporal point processes. In this paper we describe space-time point processes and introduce the package **stpp** to new users.

Keywords: epidemiology, inhomogeneous point patterns, spatial statistics, space-time point processes.

1. Introduction

A spatial point pattern is a set of data taking the form of a set of locations, irregularly distributed within a study region S, at which events have been recorded, for example the locations of trees in a naturally regenerated forest (Diggle 2003). An observed spatial point pattern can be modelled as a realisation of a spatial stochastic process represented by a set of random variables: $Y(S_m), S_m \subset S$, where $Y(S_m)$ is the number of events occurring in a sub-region S_m of S.

Many spatial processes of scientific interest also have a temporal component that may need to be considered when modelling the underlying phenomenon (e.g., distribution of cases for a disease or assessment of risk of air pollution). Spatio-temporal point processes, rather than purely spatial point processes, must then be considered as potential models. There is an extensive literature on the analysis of point process data in time (e.g., Cox and Isham 1980; Daley and Vere-Jones 2003) and in space (e.g., Cressie 1993; Diggle 2003; Møller and Waagepetersen 2003). Generic methods for the analysis of spatio-temporal point processes

are less well established; see for example Diggle (2006), Diggle and Gabriel (2010) and Section 6.6 of Cressie and Wikle (2011). There is, however, an extensive literature on the use of point process models in the specific field of seismology; see, for example, Zhuang, Ogata, and Jones (2002) and references therein.

When dealing with a single realisation of a point process, simulation methods offer various ways to understand and model the underlying process: goodness-of-fit tests, calculation of summary statistics, ... (see Illian, Penttinen, Stoyan, and Stoyan 2008). Simulation of spatial point processes is mainly implemented in the R packages **spatstat** (Baddeley and Turner 2005) and **splancs** (Rowlingson and Diggle 1993). A wide variety of random data-generation functions is implemented in **stpp**, which make up the lack of point process models in the spatio-temporal setting.

In this paper, we introduce the R (R Development Core Team 2012) package **stpp** which covers many of the models encountered in applications of point process methods to the study of spatio-temporal phenomena. First, in Section 2, we define spatio-temporal point processes and provide statistical tools for analysing their second-order properties. Then, in Section 3, we present models for such processes and some algorithms for their simulation.

2. Spatio-temporal point processes

The events of a spatio-temporal point process form a countable set of points, $\mathcal{P} = \{(s_i, t_i) : i = 1, 2, ...\}$, in which $s_i \in \mathbb{R}^2$ is the location and $t_i \in T \subset \mathbb{R}^+$ is the time of occurrence of the ith event. In practice, the data available for analysis are the points $(x_i, t_i) : i = 1, ..., n$ that form the partial realisation of the process restricted to a finite spatio-temporal domain of observation, $S \times T$, where typically S is a polygon and T a single closed interval.

In the following, $Y(A)$ denotes the number of events in an arbitrary region A.

2.1. First-order and second-order properties

First-order properties are described by the *intensity* of the process,

$$\lambda(s,t) = \lim_{|ds|\to 0, |dt|\to 0} \frac{\mathbb{E}\left[Y(ds,dt)\right]}{|ds||dt|},$$

where ds defines a small spatial region around the location s, $|ds|$ is its area, dt is a small interval containing the time t, $|dt|$ is the length of this interval and $Y(ds, dt)$ refers to the number of events in $ds \times dt$. Thus, informally, $\lambda(s,t)$ is the mean number of events per unit volume at the location (s,t). A process for which $\lambda(s,t) = \lambda$ for all (s,t) is called *homogeneous*.

Second-order properties describe the relationship between numbers of events in pairs of sub-regions within $S \times T$. The *second-order intensity* is defined as

$$\lambda_2\big((s_i,t_i),(s_j,t_j)\big) = \lim_{|Di|,|D_j|\to 0} \frac{\mathbb{E}\left[Y(D_i)Y(D_j)\right]}{|D_i||D_j|},$$

where $D_i = ds_i \times dt_i$ and $D_j = ds_j \times dt_j$ are small cylinders containing the points (s_i, t_i) and (s_j, t_j) respectively.

Other, essentially equivalent, descriptors of second-order properties include the covariance density,

$$\gamma\big((s_i,t_i),(s_j,t_j)\big) = \lambda_2\big((s_i,t_i),(s_j,t_j)\big) - \lambda(s_i,t_i)\lambda(s_j,t_j)$$

and the radial distribution function or point-pair correlation function (Cressie 1993; Diggle 2003)

$$g\big((s_i,t_i),(s_j,t_j)\big) = \frac{\lambda_2\big((s_i,t_i),(s_j,t_j)\big)}{\lambda(s_i,t_i)\lambda(s_j,t_j)}. \tag{1}$$

The covariance density is the point process analogue of the covariance function of a real-valued stochastic process. The pair correlation function can be interpreted informally as the standardised probability density that an event occurs in each of two small volumes centred on the points (s_i,t_i) and (s_j,t_j). For a spatio-temporal Poisson process (to be defined formally in Section 3.1), the covariance density is identically zero and the pair correlation function is identically 1. Larger or smaller values than these benchmarks therefore indicate informally how much more or less likely it is that a pair of events will occur at the specified locations than in a Poisson process with the same intensity.

2.2. Stationarity

A spatio-temporal point process $\{(s,t), s \in S, t \in T\}$ is *first-order and second-order stationary*:

- *in space*, if: $\lambda(s,t) \equiv \lambda(t)$ and $\lambda_2\big((s,t),s',t)\big) = \lambda_2(s-s',t)$.
- *in time*, if: $\lambda(s,t) \equiv \lambda(s)$ and $\lambda_2\big((s,t),(s,t')\big) = \lambda_2(s,t-t')$.
- *in both space and time*, if: $\lambda(s,t) = \lambda$ and $\lambda_2\big((s,t),(s',t')\big) = \lambda_2(s-s',t-t')$.

A stationary spatio-temporal point process is also *isotropic* if $\lambda_2\big((s,t),(s',t')\big) = \lambda_2(u,v)$, where (u,v) is the spatio-temporal difference vector, $u = \|s-s'\|$ and $v = |t-t'|$.

A spatio-temporal point process is *second-order intensity reweighted stationary and isotropic* if its intensity function is bounded away from zero and its pair correlation function depends only on the spatio-temporal difference vector (u,v). Second-order intensity reweighted stationarity is defined for purely spatial point processes in Baddeley, Møller, and Waagepetersen (2000). Gabriel and Diggle (2009) provide the straightforward extension to the spatio-temporal case.

2.3. Separability

A spatio-temporal point process is *first-order separable* if its intensity $\lambda(s,t)$ can be factorised as

$$\lambda(s,t) = m(s)\mu(t), \text{ for all } (s,t) \in S \times T.$$

A stationary spatio-temporal point process is *second-order separable* if the covariance density, $\gamma(u,v) = \lambda_2(u,v) - \lambda^2$, factorises as

$$\gamma(u,v) = \gamma_s(u)\gamma_t(v).$$

Note that in general, second-order separability is implied by, but does not imply, independence of the spatial and temporal component processes. However, a Poisson process has independent components if and only if it is first-order separable.

2.4. Static and dynamic plotting of spatio-temporal point process data

The most effective form of display for a spatio-temporal point process data is an animation, repeated viewing of which may yield insights that are not evident in static displays. Nevertheless, static displays are sometimes useful summaries. The **stpp** package includes four

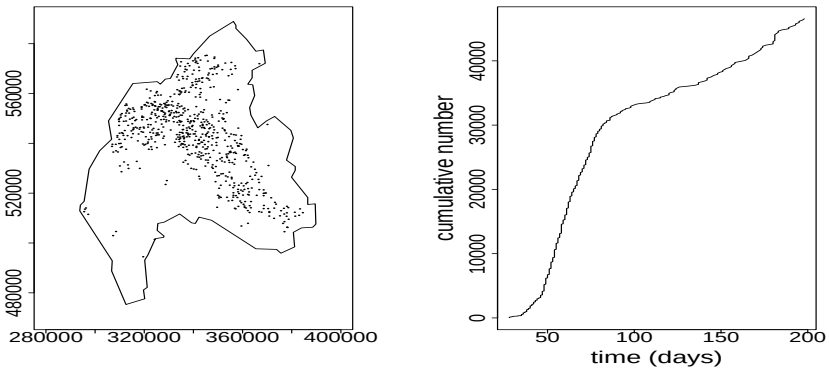


Figure 1: Static two-panel plot of data from the 2001 UK FMD epidemic in the county of Cumbria.

display functions that we illustrate using data on the locations and times (dates of reporting) of outbreaks of foot-and-mouth disease (FMD), a severe, highly communicable viral disease of farm livestock, during the UK 2001 FMD epidemic.

The data-set `fmd`, included in **stpp**, contains a three-column matrix of spatial locations and reported days (from 1 February 2001) of FMD outbreaks in the county of Cumbria. Figure 1 shows a static display of the data consisting of locations in the left-hand panel and the cumulative distribution of the times in the right-hand panel. The left-hand panel shows a very uneven distribution which, in the context of this data-set, is of limited interest without knowledge of the spatial distribution of all of the farms at risk. The right-hand panel shows the characteristic S-shape of an epidemic process. At the beginning of the epidemic the cumulative number of cases increases slowly, because the virus can be transmitted only over short distances and few of the susceptible farms are within range of the early cases. This is followed by a period of rapid increase, as the infected area spreads and there are correspondingly more susceptible farms within the transmission range. Finally, the rate of spread slows down as the epidemic is brought under control through a combination of reactive culling of infected animals and pre-emptive culling of animals at nearby farms (Keeling, Woolhouse, Shaw, Matthews, Chase-Topping, Haydon, Cornell, Kappey, Wilesmith, and Grenfell 2001).

Figure 1 can be obtained in a single command after converting the data-set into an object of class 'stpp' as follows.

```
R> library("stpp")
R> data("fmd")
R> data("northcumbria")
R> fmd <- as.3dpoints(fmd)
R> plot(fmd, s.region=northcumbria)
```

Additional graphical arguments can be passed to the plot function provided that these can be interpreted unambiguously; see `?plot.stpp` for details.

Figure 2 shows an alternative static display in which 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. This plot can be obtained as follows.

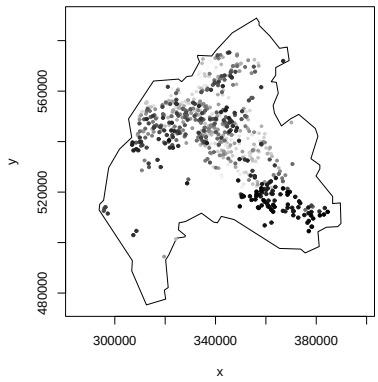


Figure 2: Static plot of data from the 2001 UK FMD epidemic. Time is treated as a quantitative mark; light grey/small dots correspond to the oldest events and dark grey/large dots correspond to the most recent events.

```
R> plot(fmd, s.region = northcumbria, pch = 19, mark = TRUE)
```

The function `animation` provides an animation of a space-time point pattern.

```
R> animation(fmd, runtime = 10, cex = 0.5, s.region = northcumbria)
```

The approximate running time of the animation (in seconds) is set through the `runtime` parameter, although the animation may actually run more slowly than this, depending on the size of the data-set and the hardware configuration. If `runtime = NULL`, the animation is displayed as quickly as the data-set and hardware configuration allow.

A second form of dynamic display is provided by the `stan` function. This enables dynamic highlighting of time slices controlled by two arguments set by using sliders: when the 'time' slider is set to T and the 'width' slider to W, highlighted points are those whose time coordinate t satisfies $T - W < t < T$. Plotting of individual locations is controlled with the '`states`' parameter. This is a list of length three specifying how locations whose associated times fall before, within and after the time window are displayed. The use of sliders allows the user to track backward or forward in time at will. This function requires the packages **rgl** (Adler and Murdoch 2012) and **rpanel** (Bowman, Gibson, Scott, and Crawford 2010), which can be downloaded and installed from the Comprehensive R Archive Network (CRAN) repository (`www.r-project.org`).

```
R> library("rgl")
R> library("rpanel")
R> stan(fmd, bgpoly = northcumbria, bgframe = FALSE)
```

Repeated viewing of either of the dynamic graphical displays shows two main features of the epidemic. Firstly, there was a progressive movement of the epidemic's focus from its origin in the north of the county to the west, and later to the south-east. Secondly, the pattern consists predominantly of spatio-temporal spread between neighbouring farms, but

with occasional and apparently spontaneous infections occurring remotely from previously infected areas.

2.5. Analysing space-time point process data

Second-order properties described in Section 2.1 are used to analyse the spatio-temporal structure of a point process. In particular, the space-time inhomogeneous pair correlation function and K-function can be used as measure of spatio-temporal clustering/regularity and as measure of spatio-temporal interaction (Gabriel and Diggle 2009; Møller and Ghorbani 2012).

Space-time inhomogeneous K-function

For a second-order intensity reweighted stationary, isotropic spatio-temporal point process, the space-time inhomogeneous K-function (STIK-function) defined by Gabriel and Diggle (2009) is

$$K_{ST}(u,v) = 2\pi \int_0^v \int_0^u g(u',v')u' \,\mathrm{d}u' \,\mathrm{d}v', \tag{2}$$

where $g(u,v) = \lambda_2(u,v)/\left(\lambda(s,t)\lambda(s',t')\right)$, $u = \|s - s'\|$ and $v = |t - t'|$. Gabriel and Diggle (2009) also give a second definition that considers both past and future events,

$$K^*_{ST}(u,v) = 2\pi \int_{-v}^v \int_0^u g(u',v')u' \,\mathrm{d}u' \,\mathrm{d}v'. \tag{3}$$

The STIK function characterizes the second-order properties of a second-order intensity reweighted stationary spatio-temporal point process, and can be used as a measure of spatio-temporal aggregation or regularity. For any inhomogeneous spatio-temporal Poisson process (see Section 3.1) with intensity bounded away from zero, $K_{ST}(u,v) = \pi u^2 v$. Values of $K_{ST}(u,v)$ greater than $\pi u^2 v$ indicate aggregation at cumulative spatial and temporal separations less than u and v, whilst $K_{ST}(u,v) < \pi u^2 v$ indicates regularity. The STIK function can also be used to test for space-time clustering and space-time interaction (Gabriel and Diggle 2009; Møller and Ghorbani 2012).

The function `STIKhat` implements a non-parametric estimator of the STIK function, as defined by

$$\widehat{K}_{ST}(u,v) = \frac{1}{|S \times T|} \frac{n}{n_v} \sum_{i=1}^{n_v} \sum_{j=1;j>i}^{n_v} \frac{1}{w_{ij}} \frac{1}{\lambda(s_i,t_i)\lambda(s_j,t_j)} \mathbf{1}_{\{\|s_i - s_j\| \le u \ ; \ t_j - t_i \le v\}}. \tag{4}$$

if parameter `infectious = TRUE` or by

$$\widehat{K}^*_{ST}(u,v) = \frac{1}{|S \times T|} \sum_{i=1}^{n} \sum_{j \ne i} \frac{1}{w_{ij} v_{ij}} \frac{1}{\lambda(s_i,t_i)\lambda(s_j,t_j)} \mathbf{1}_{\{\|s_i - s_j\| \le u \ ; \ |t_j - t_i| \le v\}}. \tag{5}$$

otherwise. In Equation 4, n_v is the number of events for which $t_i \le T_1 - v$, $T = [T_0, T_1]$. In Equations 4 and 5, w_{ij} denotes the Ripley's spatial edge correction factor. This consists in weighting by the proportion of the circumference of a circle centred at the location s_i with radius $\|s_i - s_j\|$ lying in S. In Equation 5 v_{ij} denotes the temporal edge correction factor (the one-dimensional analogue of the Ripley's edge correction factor). It is equal to 1 if both ends of the interval of length $2|t_i - t_j|$ centred at t_i lie within T and 1/2 otherwise.

In practice, $\lambda(x)$ must be estimated. See Gabriel (2012) for a discussion of such an estimation.

Space-time inhomogeneous pair correlation function

An estimator of the space-time pair correlation function defined in Equation 1 is

$$\hat{g}(u,v) = \frac{1}{|S \times T|} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1}{w_{ij} v_{ij}} \frac{k_s(u - \|s_i - s_j\|) k_t(v - |t_i - t_j|)}{\lambda(s_i, t_i) \lambda(s_j, t_j)},$$

where w_{ij} and v_{ij} are the spatial and temporal edge correction factors defined in Equation 5 and $k_s(\cdot)$, $k_t(\cdot)$ are kernel functions with bandwidths h_s and h_t. Experience with pair correlation function estimation recommends box kernels, see Illian *et al.* (2008).

Application to FMD data

The functions `STIKhat` and `PCFhat` provide estimates of the space-time inhomogeneous K-function and pair correlation function. The following code applies these estimators to the FMD data under the assumption that the spatio-temporal intensity is separable. The spatial intensity is estimated using the function `kernel2d` of the package **splancs**. Other R packages capable of this type of estimation include **spatstat** and **spatialkernel** (Zheng and Diggle 2012). In `PCFhat` the box kernel is used by default. 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** (Wand and Ripley 2012). Note that the bandwidths play an important role in determining the quality of the estimators as they heavily influence the trade-off between bias and variance.

```
R> FMD <- as.3dpoints(fmd[, 1] / 1000, fmd[, 2] / 1000, fmd[,3])
R> Northcumbria <- northcumbria / 1000

R> Mt <- density(FMD[ ,3], n = 1000)
R> mut <- Mt$y[findInterval(FMD[ ,3], Mt$x)] * dim(FMD)[1]

R> h <- mse2d(as.points(FMD[,1:2]), Northcumbria, nsmse = 50, range = 4)
R> h <- h$h[which.min(h$mse)]
R> Ms <- kernel2d(as.points(FMD[ ,1:2]), Northcumbria, h = h, nx = 5000,
+  ny = 5000)
R> atx <- findInterval(x = FMD[ ,1], vec = Ms$x)
R> aty <- findInterval(x = FMD[ ,2], vec = Ms$y)
R> mhat <- NULL
R> for(i in 1:length(atx)) mhat <- c(mhat, Ms$z[atx[i],aty[i]])

R> u <- seq(0, 10, by = 1)
R> v <- seq(0, 15, by = 1)
R> stik <- STIKhat(xyt = FMD, s.region = Northcumbria, t.region = c(1, 200),
+  lambda = mhat * mut / dim(FMD)[1], dist = u, times = v, infectious = TRUE)

R> g <- PCFhat(xyt = FMD, lambda = mhat * mut / dim(FMD)[1], dist = 1:20,
+  times = 1:20, s.region = Northcumbria, t.region = c(1,200))
```

We can plot the estimates by using the functions `plotK` and `plotPCF` which provide either a contour plot (default) or a perspective plot (when `persp=TRUE`).

```
R> plotK(stik)
R> plotPCF(g)
R> plotPCF(g, persp = TRUE, theta = -65, phi = 35)
```

Figure 3 shows such plots for the FMD data, where u denotes distances in kilometers and v times in days. To assess the data for evidence of spatio-temporal clustering, we can follow

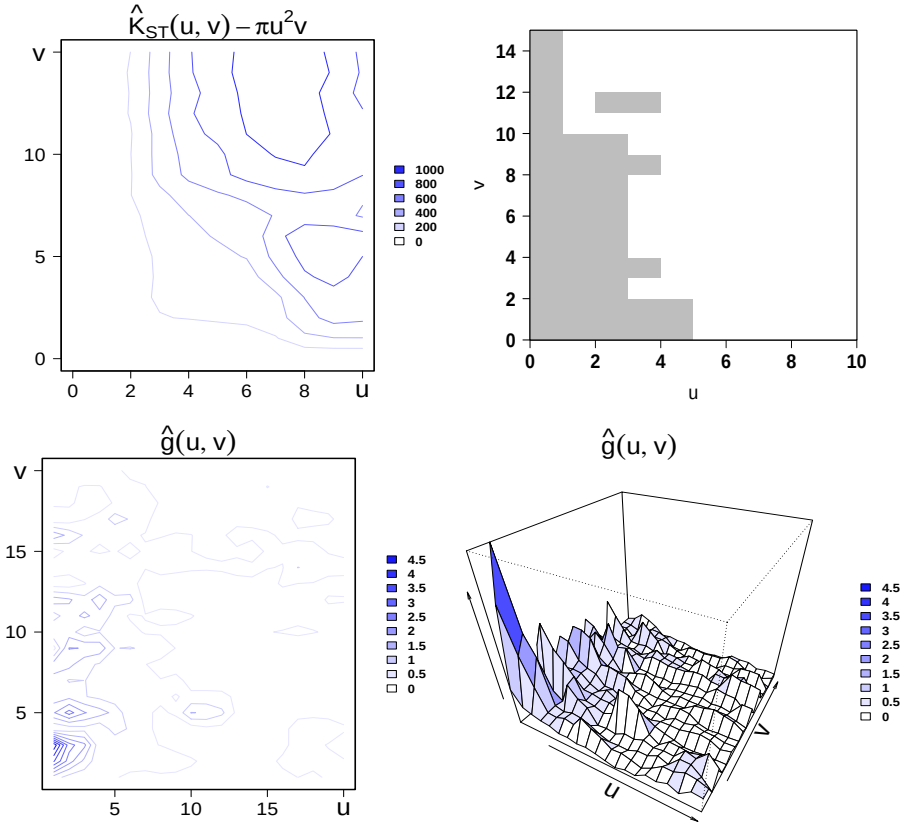


Figure 3: Contour plot (left) and perspective plot (bottom right) of the STIK function and (top) pair correlation function (bottom) estimated from FMD data. Comparison between $\widehat{K}(u,v) - \pi u^2 v$ and tolerance envelopes indicating spatio-temporal clustering in grey shading (top right).

common practice by comparing the estimator $\widehat{K}(u,v)$ with estimates calculated for simulations under the null hypothesis that the underlying process is an inhomogeneous Poisson process (see Section 3.1). We then compare the data with simulations of a Poisson process with intensity $\hat{\lambda}(s,t) = m(s)\mu(t)$. The top right panel of Figure 3 shows comparison between $\widehat{K}(u,v) - \pi u^2 v$ and tolerance envelopes indicating spatio-temporal clustering (grey shading). It indicates spatio-temporal clustering at small temporal distances $v < 10$ days and spatial distances $u < 5$ kilometers. This corresponds to the contour plot of the pair correlation

function in Figure 3, where values greater than one indicate clustering. The FMD data-set has been further analysed in Møller and Ghorbani (2012).

3. Models

Space-time point pattern data are increasingly available in a wide range of scientific settings. Data-sets of this kind usually consist of a single realisation of the underlying process. Usually, separate analyses of the spatial and the temporal components are of limited value, because the scientific objectives of the analysis are to understand and to model the underlying spatio-temporally interacting stochastic mechanisms. Simulation of spatio-temporal point processes is a useful tool, both for understanding the behaviour of models and as necessary component of Monte Carlo methods of inference. Packages that deal with spatio-temporal data include **gstat** (Pebesma 2004) for geostatistical data, **spacetime** (Pebesma, Graeler, and Gottfried 2012) for lattice data, **splancs** and **spatstat** for point process data. However, neither **splancs** nor **spatstat** includes functions for simulating spatio-temporal data. The **lgcp** package (Taylor, Davies, Rowlingson, and Diggle 2012) does include functions for simulating log-Gaussian Cox processes (Møller, Syversveen, and Waagepetersen 1998), but its focus is on methods of inference within this model-class. In contrast, **stpp** focuses on simulation over a wide class of models.

Models and functions are described below. All functions return a matrix `xyt` (or list of matrices if the number of simulations, `nsim`, is greater than 1) containing the points (x, y, t) of the simulated point pattern and `s.region`, `t.region` which are the spatial and temporal regions passed in argument. By default, the spatio-temporal region is the unit cube. The spatial region can be a polygon defined by a two-columns matrix in `s.region`. Note that `xyt` (or any element of the list if `nsim`>1) is an object of the class 'stpp'.

3.1. Poisson process

Homogeneous Poisson process

The homogeneous Poisson process is the simplest possible stochastic mechanism for the generation of spatio-temporal point patterns. It is rarely plausible as a model for data, but provides a benchmark of complete spatio-temporal randomness (CSTR). Informally, in a realisation of a homogenous Poisson process on any spatio-temporal region $S \times T$, the events form an independent random sample from the uniform distribution on $S \times T$. More formally, the homogeneous Poisson process is defined by the following postulates:

1. For some $\lambda > 0$, the number $Y(S \times T)$ of events within the region $S \times T$ follows a Poisson distribution with mean $\lambda|S||T|$, where $|\cdot|$ denotes (two-dimensional) area or (one-dimensional) length according to context.

2. Given $Y(S \times T) = n$, the n events in $S \times T$ form an independent random sample from the uniform distribution on $S \times T$.

The first-order and second-order intensities of a homogeneous Poisson process reduce to constants, $\lambda(s, t) = \lambda$ and $\lambda_2\big((s_i, t_i), (s_j, t_j)\big) = \lambda^2$. Hence, as stated in Section 2.1, the covariance

density is identically zero, the pair correlation function identically 1, and the STIK function is $K_{ST}(u, v) = \pi u^2 v$.

Simulation

To generate a homogeneous Poisson point pattern in $S \times T$, **stpp** uses a two-step procedure:

1. *Simulate the number of events $n = Y(S \times T)$ occurring in $S \times T$ according to a Poisson distribution with mean $\lambda|S||T|$.*
2. *Sample each of the n locations and n times according to a uniform distribution on S and on T respectively.*

Inhomogeneous Poisson process

The inhomogeneous Poisson process is the simplest non-stationary point process. It is obtained replacing the constant intensity λ of a homogeneous Poisson process by a spatially and/or temporally varying intensity function $\lambda(s, t)$. Inhomogeneous Poisson processes are defined by the following postulates:

1. The number $Y(S \times T)$ of events within the region $S \times T$ follows a Poisson distribution with mean $\int_S \int_T \lambda(s, t)\, \mathrm{d}t\, \mathrm{d}s$.
2. Given $Y(S \times T) = n$, the n events in $S \times T$ form an independent random sample from the distribution on $S \times T$ with probability density function $f(s, t) = \lambda(s, t) / \int_S \int_T \lambda(\tilde{s}, \tilde{t})\, \mathrm{d}\tilde{t}\, \mathrm{d}\tilde{s}$.

For a Poisson process with intensity $\lambda(s, t)$, the second-order intensity is $\lambda_2\big((s_i, t_i), (s_j, t_j)\big) = \lambda(s_i, t_i)\lambda(s_j, t_j)$, hence the covariance density is identically zero, the pair correlation function identically 1, and the STIK function $K_{ST}(u, v) = \pi u^2 v$ as in the homogeneous case.

Simulation

To generate a realisation of an inhomogeneous Poisson process in $S \times T$, **stpp** uses a thinning algorithm as follows. For a given intensity function $\lambda(s, t)$:

1. *Define an upper bound λ_{max} for the intensity function $\lambda(s, t)$.*
2. *Simulate a homogeneous Poisson process with intensity λ_{max}.*
3. *"Thin" the simulated process as follows,*
   (a) *Compute $p = \lambda(s, t)/\lambda_{max}$ for each point (s, t) of the homogeneous Poisson process.*
   (b) *Generate a sample u from the uniform distribution on $(0, 1)$.*
   (c) *Retain the locations for which $u \leq p$.*

Examples

Poisson processes are simulated by the function `rpp`. Realisations are simulated in a region $S \times T$, where S is a polygon and T is an interval, with default the unit cube. For a homogeneous Poisson process, the intensity is specified by a constant. For example, the sequence of commands

```
R> hpp1 <- rpp(lambda = 200, nsim = 5, replace = FALSE)
R> stan(hpp1$xyt[[2]])
```

generates five realisations of the Poisson process with intensity $\lambda = 200$ in the unit cube and displays the second realisation dynamically.

The sequence of commands

```
R> data("northcumbria")
R> hpp2 <- rpp(npoints = 1000, s.region = northcumbria, t.region = c(1, 500),
+  discrete.time = TRUE)
R> animation(hpp2$xyt, s.region = hpp2$s.region)
```

generates and displays a realisation of the Poisson process with intensity $\lambda = 1000/(|S||T|)$, but conditioned to produce exactly 1000 points in the region $S \times T$, where S is the county of Cumbria and $T = [1, 500]$. The argument `npoints` specifies the fixed value of the number of events to be generated. Simulated times are restricted to integers (set by the `discrete.time` parameter), and coincident times are allowed (set by the `replace` argument).

Figure 4 illustrates through a static display realisations of "`hpp1`" (left) and "`hpp2`" (right) defined above. Here, time is treated as a quantitative mark; light grey/small dots correspond to the oldest events and dark grey/large dots correspond to the most recent events. In the following, we shall use such plot to illustrate the realisations of spatio-temporal point processes.

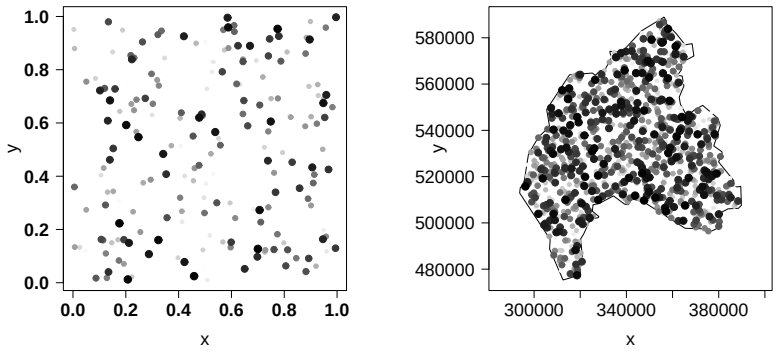


Figure 4: Realisations of the homogeneous Poisson processes "`hpp1`" (left) and "`hpp2`" (right) defined in examples.

The function `rpp` can also generate realisations of inhomogeneous Poisson processes. The intensity is then specified either by a function of the coordinates and times, $\lambda(x, y, t, ...)$, or by a three dimensional array. For example, the sequence of commands

```
R> lbda1 <- function(x, y, t, a){a * exp(-4 * y) * exp(-2 * t)}
R> ipp1 <- rpp(lambda = lbda1, npoints = 200,
+  a = 1600 / ((1 - exp(-4)) * (1 - exp(-2))))
R> stan(ipp1$xyt)
```

generates 200 points of the Poisson process with intensity $\lambda(x, y, t) = ae^{-4y-2t}$ in the unit cube. The constant $a = 1600/\{(1 - e^{-4})(1 - e^{-2})\}$ ensures that the mean number of points is 200. When the `npoints` argument is omitted, the number of points is not fixed by the user but is generated by a Poisson distribution with mean $\iint_S \int_T \lambda(x, y, t, ...)\, \mathrm{d}t\, \mathrm{d}x\, \mathrm{d}y$. Realisations can also be generated when the intensity is specified by a spatio-temporal intensity array. In the following example, we estimate the spatial and temporal intensities of the `fmd` data by kernel smoothing (Silverman 1986; Berman and Diggle 1989) and display the realisation superimposed on a grey-scale image of the spatial intensity estimate. Figure 5 illustrates the estimate of the spatial (left) and temporal (right) intensity functions. Dark/light grey correspond to high/weak values of the spatial intensity.

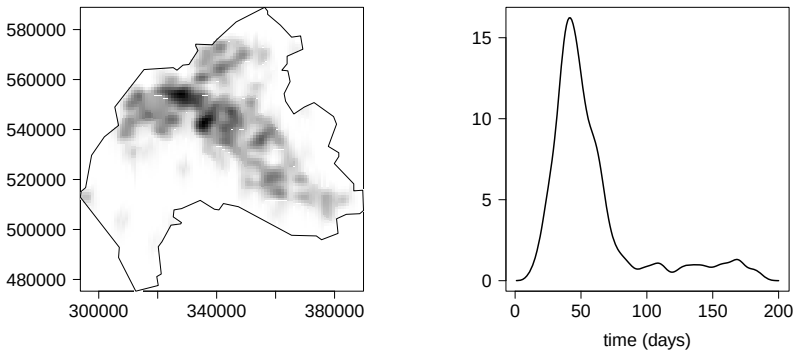


Figure 5: Spatial (left) and temporal (right) intensity functions estimated from the `fmd` dataset.

```
R> data("fmd")
R> data("northcumbria")
R> h <- mse2d(as.points(fmd[, 1:2]), northcumbria, nsmse = 30, range = 3000)
R> h <- h$h[which.min(h$mse)]
R> Ls <- kernel2d(as.points(fmd[, 1:2]), northcumbria, h, nx = 100, ny = 100)
R> Lt <- dim(fmd)[1] * density(fmd[, 3], n = 200)$y
R> Lst <- array(0, dim = c(100, 100, 200))
R> for(k in 1:200) Lst[,,k] <- Ls$z * Lt[k] / dim(fmd)[1]
R> ipp2 <- rpp(lambda = Lst, s.region = northcumbria, t.region = c(1, 200),
+  discrete.time = TRUE)
R> image(Ls$x, Ls$y, Ls$z, col = grey((1000:1) / 1000))
R> polygon(northcumbria)
R> animation(ipp2$xyt, add = TRUE, cex = 0.5, runtime = 15)
```

Figure 6 illustrates realisations of "`ipp1`" (left) and "`ipp2`" (right) defined above.

3.2. Poisson cluster process

We define a spatio-temporal Poisson cluster process as the following direct generalization of its spatial counterpart (Neyman and Scott 1958).

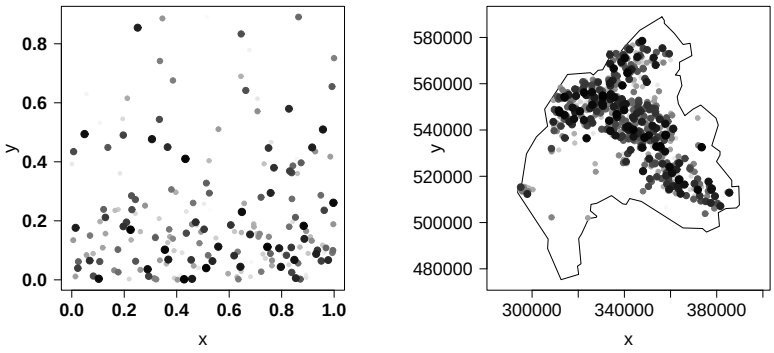


Figure 6: Realisations of the inhomogeneous Poisson processes "`ipp1`" (left) and "`ipp2`" (right) defined in examples.

1. Parents form a Poisson process with intensity $\lambda_p(s,t)$.
2. The number of offspring per parent is a random variable N_c with mean m_c, realised independently for each parent.
3. The positions and times of the offspring relative to their parents are independently and identically distributed according to a trivariate probability density function $f(\cdot)$ on $\mathbb{R}^2 \times \mathbb{R}^+$.
4. The final process is composed of the superposition of the offspring only.

Simulation

To generate a Poisson cluster point process in $S \times T$ we use the following three-step procedure:

1. *Simulate a Poisson process of parent points with intensity $\lambda_p(s,t)$ in $S' \times T'$, where $S' \supset S$ and $T' \subset T$ so as to avoid, or at least minimise, edge-effects that would otherwise result from the loss of offspring from parents close to the boundary of S and T.*
2. *For each simulated parent, generate a random number n_c of offspring from a Poisson distribution with mean m_c.*
3. *Generate the spatio-temporal displacements of the offspring from their parents as independent realisations from the trivariate distribution with density $f(\cdot)$.*

Examples

The function `rpcp` generates points around a number of parents points generated by `rpp`. Their spatial and temporal distributions can be chosen among "uniform", "normal" and "exponential" using the `cluster` argument. This can be either a single value if the distribution in space and time is the same, or a vector of length two, giving first the spatial distribution of offspring relative to their parents and then the temporal distribution. The parameter `dispersion` is a scale parameter, equals to twice the standard deviation of location of children

relative to their parents for a normal distribution of children, the mean for an exponential distribution and half the range for a uniform distribution. By default, `edge = "larger.region"`. The function generates the Poisson cluster process within a larger region but return only those points that fall within $S \times T$. If `edge = "without"` the process is generated only in $S \times T$ and will have an artificially reduced intensity near the boundary of S. By default, the larger spatial region is the convex hull of `s.region` enlarged by the spatial element of `dispersion` and the larger time interval is `t.region` enlarged by the temporal element of `dispersion`. The user can over-ride the default using the two-element vector argument `larger.region`.

In the following example, parents are generated by a homogeneous spatio-temporal Poisson process with intensity $\lambda = n_p/(|S||T|)$, where S is the boundary of Cumbria, $T = [1, 365]$ and $n_p = 50$ is the number of parents. Each parent gives birth to a series of offspring; the number of offspring per parent follows a Poisson distribution with mean `mc`.

```
R> data("northcumbria")
R> pcp1 <- rpcp(nparents = 50, mc = 10, s.region = northcumbria,
+  t.region = c(1, 365), cluster = c("normal", "exponential"),
+  dispersion = c(5000, 5))
R> animation(pcp1$xyt, s.region = pcp1$s.region, t.region = pcp1$t.region,
+  runtime = 5)
```

The sequence of commands

```
R> lbda <- function(x, y, t, a){a * exp(-4 * y) * exp(-2 * t)}
R> pcp2 <- rpcp(nparents = 50, npoints = 250, cluster = "normal",
+  lambda = lbda, a = 2000 / ((1 - exp(-4)) * (1 - exp(-2))))
R> stan(pcp2$xyt)
```

generates a realisation of the Poisson cluster process in the unit cube and displays the realisation. Here, the parent process is Poisson with intensity $\lambda(x, y, t) = ae^{-4y-2t}$ and the offspring are normally dispersed both in space and in time. The constant $a = 2000/\{(1-e-4)(1-e-2)\}$ has been chosen so that the mean number of points is 250. Figure 7 illustrates realisations of "`pcp1`" (left) and "`pcp2`" (right) defined above.

3.3. Interaction processes

Inhibition process

Inhibition processes either prevent (strict inhibition) or make unlikely the occurrence of pairs of close events, resulting in patterns that are more regular in space and/or in time than a Poisson process of the same intensity.

In a spatial simple sequential inhibition process (strict inhibition), let δ_s denote the minimum permissible distance between events and λ_s the spatial intensity of the process. The proportion of the plane covered by non-overlapping discs of radius $\delta_s/2$ is $\rho = \lambda_s \pi \delta_s^2/4$, which we call the packing density. The maximum achieveable packing density is for a pattern of points in a regular triangular lattice at spacing δ_s, for which $\rho = \sqrt{3}/2 \approx 0.87$. Depending on exactly how the points are generated, even this value of δ_s may not be feasible; for example,

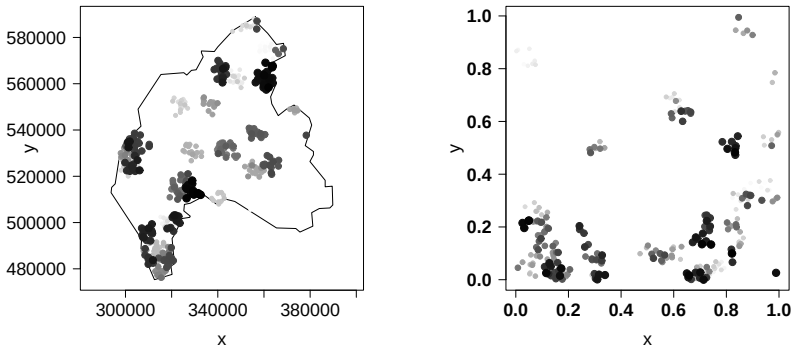


Figure 7: Realisations of the Poisson cluster processes "pcp1" (left) and "pcp2" (right) defined in examples.

if the points are placed sequentially at random in a large region S, the maximum achieveable packing density is approximately 0.55 (Tanemura 1979).

Simple sequential inhibition processes in space and time are defined by the following algorithm. Consider a sequence of m events (s_i, t_i) in $S \times T$. Then,

1. s_1 and t_1 are uniformly distributed in S and T respectively.
2. At the kth step of the algorithm, $k = 2, ..., m$, s_k is uniformly distributed on the intersection of S with $\{s : \|s - s_j\| \geq \delta_s, j = 1, \ldots, k-1\}$ and t_k is uniformly distributed on the intersection of T with $\{t : |t - t_j| \geq \delta_t, j = 1, \ldots, k-1\}$.

To obtain a larger class of inhibition processes than the one defined above, we extend condition 2. of the above algorithmic definition by introducing functions $p_s(u)$ and $p_t(v)$ that together determine the probability that a potential point at location s and time t will be accepted as a point of the process, according to the following algorithm, in which the functions $g_s(\cdot)$, $g_t(\cdot)$, $h_s(\cdot)$, $h_t(\cdot)$ and the parameter r are to be defined.

1. s_1 and t_1 are uniformly distributed in S and T respectively.
2. At the kth step of the algorithm, $k = 2, ..., m$,
   (a) Generate uniformly a location $s \in S$ and a time $t \in T$.
   (b) Generate $u_s \sim \mathcal{U}[0, 1]$ and $u_t \sim \mathcal{U}[0, 1]$.
   (c) If $\|s - s_j\| \geq \delta_s$ for all $j = 1, \ldots, k-1$, then set $p_s = 1$.
   Otherwise compute $p_s = g_s\left(h_s\left((\|s - s_j\|)_{j=1,\ldots,k-1}, \theta_s, \delta_s\right), r\right)$.
   (d) If $|t - t_j| \geq \delta_t$ for all $j = 1, \ldots, k-1$, then set $p_t = 1$.
   Otherwise compute $p_t = g_t\left(h_t\left((|t - t_j|)_{j=1,\ldots,k-1}, \theta_t, \delta_t\right), r\right)$.
   (e) If $u_s < p_s$ and $u_t < p_t$, then keep s and t.

Within the **stpp** package, the functions g_s and g_t can be chosen among "min", "max" and "prod". This allows us to consider either the minimum or the maximum or the product

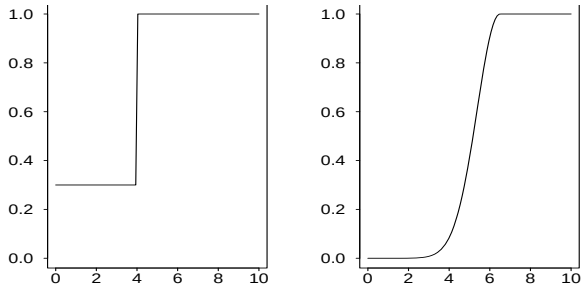


Figure 8: Inhibition functions implemented in `rinter`; *left*: step, $\delta = 4$, $\theta = 0.3$; *right*: Gaussian, $\delta = 2$, $\theta = 9$.

of terms $h_s(\cdot)$ and $h_t(\cdot)$ from all previous events or only the r most recent in time (set by `recent` argument). For example, setting $g_s =$"prod" and $r \leq k-1$, at the kth step of the algorithm, the probability of acceptance is $p_s = \prod_{j=k-r}^{k-1} h_s\left(\|s - s_j\|, \theta_s, \delta_s\right)$. The functions h_s and h_t define the nature of the interaction between a pair of points according to their spatial and temporal separations, respectively. In the following, $h(\cdot)$ and δ without subscripts can represent either $h_s(\cdot)$ or $h_t(\cdot)$ and δ_s or δ_t. The function $h(\cdot)$ is monotone, increasing, tends to 1 when the separation tends to infinity and satisfies $0 \leq h(\cdot) \leq 1$. Currently the following functions are implemented in **stpp**:

- step: $h(x) = \begin{cases} 1, & \text{if } x > \delta \\ \theta, & \text{otherwise} \end{cases}$, $\theta \in [0, 1]$.
- Gaussian: $h(x) = \begin{cases} 1, & \text{if } x > \delta + \theta/2 \\ \exp\left\{-\frac{(x-\delta-\theta/2)^2}{2(\theta/8)^2}\right\}, & \text{if } \delta < x \leq \delta + \theta/2 \\ 0, & \text{if } x \leq \delta \end{cases}$, $\theta \geq 0$.

One distinction between these two functions is that the 'step' function allows points to be generated at distances less or equal to δ if $\theta > 0$, whereas the 'Gaussian' function does not. Figure 8 gives an example of each.

Contagious process

A simple contagious processes in space and time can be defined algorithmically as follows. Consider a sequence of m events (s_i, t_i) in $S \times T$. Then,

1. s_1 and t_1 are uniformly distributed in S and T respectively.
2. At the kth step of the algorithm, given $\{(s_j, t_j), j = 1, \ldots, k-1\}$, s_k is uniformly distributed on the intersection of S and the circle of center s_{k-1} and radius δ_s, whilst t_k is uniformly distributed on the intersection of T and the segment $[t_{k-1}, t_{k-1} + \delta_t]$.

As in the case of inhibitory processes, we enlarge the class of contagious processes by introducing functions p_s and p_t, which depend on $\|s - s_j\|$ and $|t - t_j|$ respectively. The kth step of this algorithm is

1. Generate uniformly a location $s \in S$ and a time $t \in T$.

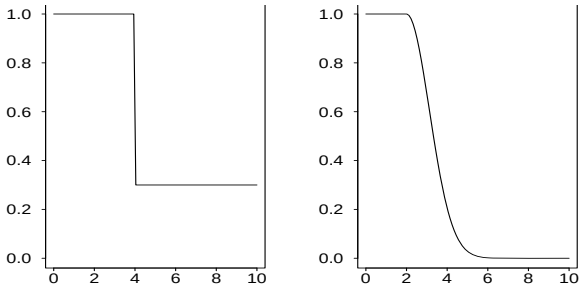


Figure 9: Contagious functions implemented in `rinter`; *left*: step, $\delta = 4$, $\theta = 0.3$; *right*: Gaussian, $\delta = 2$, $\theta = 9$.

2. Generate $u_s \sim \mathcal{U}[0,1]$ and $u_t \sim \mathcal{U}[0,1]$.
3. If $\|s - s_j\| < \delta_s$ for all $j = 1, \ldots, k-1$, then set $p_s = 1$.
   Otherwise compute $p_s = g_s\left(h_s\left(\left(\|s - s_j\|\right)_{j=1,\ldots,k-1}, \theta_s, \delta_s\right), r\right)$.
4. If $|t - t_j| < \delta_t$ for all $j = 1, \ldots, k-1$, then set $p_t = 1$.
   Otherwise compute $p_t = g_t\left(h_t\left(\left(|t - t_j|\right)_{j=1,\ldots,k-1}, \theta_t, \delta_t\right), r\right)$.
5. If $u_s < p_s$ and $u_t < p_t$, then keep s and t.

The g and h functions have the same interpretation as for inhibitory processes. The same set of three g functions is currently implemented, whilst the two implemented h functions are analogous to their inhibitory counterparts:

- step: $h(x) = \begin{cases} 1, & \text{if } x \leq \delta \\ \theta, & \text{otherwise} \end{cases}$, $\theta \in [0,1]$.
- Gaussian: $h(x) = \begin{cases} 1, & \text{if } x \leq \delta \\ \exp\left\{-\frac{(x-\delta)^2}{2(\theta/8)^2}\right\}, & \text{otherwise} \end{cases}$, $\theta \geq 0$.

Figure 9 gives an example of each.

Examples

The function `rinter` generates both inhibitory and contagious processes, differentiated by the parameter `inhibition`. The parameter `recent` allows the user to consider either all or only the r most recent events.

The simple sequential inhibition process is obtained by choosing "min" for both g functions, "step" for both h functions and 0 for the θ parameter. The commands

```
R> inh1 <- rinter(npoints = 200, thetas = 0, deltas = 0.05, thetat = 0,
+  deltat = 0.001, inhibition = TRUE)
R> stan(inh1$xyt)
```

generate one realisation of this process in the unit cube and display the realisation.

Similarly, the commands

```
R> data("northcumbria")
R> cont1 <- rinter(npoints = 250, s.region = northcumbria,
+  t.region = c(1, 200), thetas = 0, deltas = 7500, thetat = 0, deltat = 10,
+  recent = 1, inhibition = FALSE)
R> plot(cont1$xyt, pch = 19, s.region = cont1$s.region, mark = TRUE,
+  mark.col = 4)
R> animation(cont1$xyt, s.region = cont1$s.region, t.region = cont1$t.region,
+  incident = "red", prevalent = "lightgreen", runtime = 15, cex = 0.8)
```

generate one realisation of the simple contagious process in a prescribed, irregular spatio-temporal region and display the realisation.

The simple contagious process is similarly specified by using "step" for both h functions, 0 for the θ parameter and $r = 1$.

The user can also call their own functions h_s and h_t, which can combine inhibitory and contagious elements provided that the functions only depend on d (spatial or temporal separation between two points) and two parameters, θ and δ. Whilst the user is allowed to define their own functions, he has to remind that all conditions on $h(\cdot)$ must be satisfied: $h(\cdot)$ is monotone, increasing, tends to 1 when d tends to infinity and satisfies $0 \leq h(\cdot) \leq 1$. This is illustrated in the following example, with $h_s(\cdot)$ and $ht(\cdot)$ plotted in the bottom left panel of Figure 10.

```
R> 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)}

R> 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)}

R> d <- seq(0, 1, length = 100)
R> plot(d, hs(d, 0.2, 0.1, 0.1), xlab = "", ylab = "", type = "l",
+  ylim = c(0,1), lwd = 2, las = 1)
```

```
R> lines(d, ht(d, 0.1, 0.05, 0.3), col = 2, lwd = 2)
R> legend("bottomright", col = 1:2, lty = 1, lwd = 2, bty = "n", cex = 2,
+  legend = c(expression(h[s]), expression(h[t])))

R> 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)
R> animation(inh2$xyt, runtime = 15, cex = 0.8)
```

Figure 10 illustrates realisations of the interaction processes: simple inhibition process (top left), contagious process (top right), inhibition process with interaction functions defined by the user (bottom right).

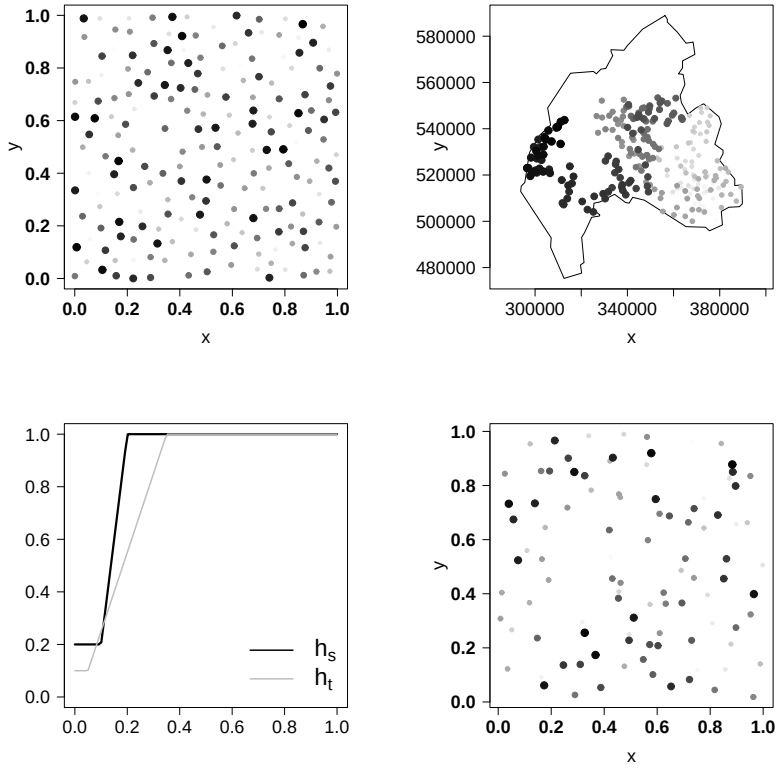


Figure 10: Realisations of the interaction processes "`inh1`" (top left), "`cont1`" (top right) and "`inh2`" (bottom right) with interaction function defined by the user (bottom left).

3.4. Infectious processes

The difference between an infectious and a contagious disease is that the former can be contracted by a person without their having come into direct contact with an infected person, whilst the latter is transmitted only by direct contact. All contagious diseases are infectious, but many infectious diseases are not contagious. Here, we use the term *contagious process* to mean that the existence of a point of the process at location s and time t increases the

likelihood of there being additional points of the process close to (s, t) in both space and time. We use the term *infectious process* in a narrower sense, whereby to each infected individual at a time t there corresponds an infection rate $h(t)$, which we here assume depends on three parameters: a latent period α, the maximum infection rate β and the infection period γ. Note that an infectious process in this sense may exhibit a combination of contagious and inhibitory properties. Diggle, Kaimi, and Abellana (2010) give an example of such a process to describe the pattern of colonisation of a nesting ground, in which each new arrival tends to choose a nesting location close to established nests, but not so close as to invade their established territories.

We define an infectious process in space and time as follows. Consider a sequence of m events $\{(s_i, t_i), i = 1, \ldots, m\}$ in $S \times T$. Then,

1. Choose the location s_1 and time t_1 of the first event.
2. Given $\{(s_j, t_j), j = 1, \ldots, k-1\}$, s_k is either radially symmetrically distributed around s_{k-1} or is a point in a Poisson process with intensity $\lambda(s)$, and t_k is either uniformly or exponentially distributed from t_{k-1}. We denote by f_s and f_t the distribution of s_k and t_k relative to s_{k-1} and t_{k-1}, respectively.

Simulation

The algorithm used in the **stpp** package to simulate an infectious processes is as follows.

Step 1

1. Set (s_1, t_1).

Step $k = 2, 3, ...$

2. Compute $h_k(t) = h(t|t_{k-1}, \alpha, \beta, \gamma)/\{\int_T h(u)\,\mathrm{d}u\}$ and $\mu_k(t) = \sum_{j=1}^{k} h_j(t)$.
3. Generate $u_k \sim \mathcal{U}[0, 1]$.
4. Generate $t_k = t_{k-1} + v$ and $s_k = s_{k-1} + w$, where v and w are generated from f_t and f_s, respectively.
5. If $\begin{cases} \|s_k - s_j\| \geq \delta_s, & \text{for an inhibition process} \\ \|s_k - s_j\| < \delta_s, & \text{for a contagious process} \end{cases}$, $j = 1, \ldots, k-1$, then set $p_k = 1$.

   Otherwise, compute $p_k = g\big(\{\mu_j(t_1, \ldots, t_j)/\max_j \mu_j(t_1, \ldots, t_j)\}_{j=1,\ldots,k}, r\big)$.
6. If $u_k < p_k$, then keep (s_k, t_k). Otherwise, generate another candidate.

As before, the function g can be chosen amongst "min", "max" and "prod" and is computed from either all previous events or the r most recent in time. The spatial distribution f_s can be chosen among:

- uniform: $s_k = (x_k, y_k)$, where $x_k = x_{k-1} + \mathcal{U}[-d_s, d_s]$ and $y_k = y_{k-1} + \mathcal{U}[-d_s, d_s]$.
- Gaussian: $s_k = (x_k, y_k)$, where $x_k = x_{k-1} + \mathcal{N}(0, d_s/2)$ and $y_k = y_{k-1} + \mathcal{N}(0, d_s/2)$.
- exponential: $s_k = (x_k, y_k)$, where $x_k = x_{k-1} \pm \mathcal{E}xp(1/d_s)$ and $y_k = y_{k-1} \pm \mathcal{E}xp(1/d_s)$.

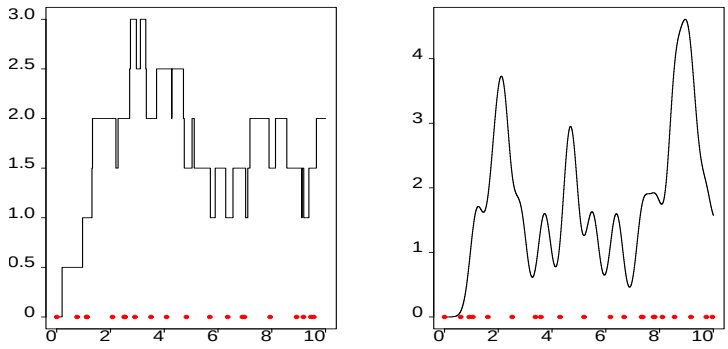


Figure 11: Illustration of $\mu_k(t)$ for h as a step function (left) and a Gaussian function (right). Solid dots indicate the times of points of the process.

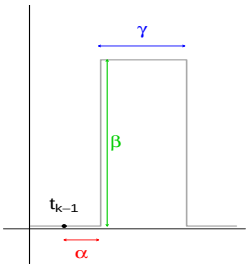
- Poisson: s_k a point in a Poisson process with intensity $\lambda(s)$.

The temporal distribution f_t can be chosen among:

- uniform: $t_k = t_{k-1} + \mathcal{U}[0, d_t]$.
- exponential: $t_k = t_{k-1} + \mathcal{E}xp(1/d_t)$.

The infection rate h depends on t_{k-1}, the latent period α, the maximum infection rate $\beta \in [0, 1]$ and the infection period γ. The options currently implemented are:

- step: $h(t) = \begin{cases} \beta, & \text{if } t_{k-1} + \alpha \leq t \leq t_{k-1} + \alpha + \gamma \\ 0, & \text{otherwise} \end{cases}$,



- Gaussian: $h(t) = \beta \exp\left\{-\big(t - (t_{k-1} + \alpha + \gamma/2)\big)^2 / 2(\gamma/8)^2\right\}$.

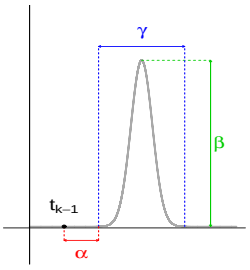


Figure 11 illustrates $\mu_k(t)$ for h as a step function (left) or a Gaussian function (right), with $\alpha = 0.2$, $\beta = 0.7$ and $\gamma = 2$ in each case. Dots correspond to the times of events.

Examples

The function `rinfec` generates infectious processes defined by an infection rate function $h(\alpha, \beta, \gamma)$ (arguments `h`, `alpha`, `beta` and `gamma`) and inhibition or contagious processes (differentiated by the argument `inhibition`). Spatial and temporal distribution are specified

through the arguments `s.distr`, `t.distr` and `maxrad`, where `maxrad` is a two-element vector defining the spatial and temporal dispersions, respectively. The spatial distribution can be specified as `uniform`, `"gaussian"`, `"exponential"` or `"poisson"`, and the temporal distribution as `"uniform"` or `"exponential"`. The probability of acceptance of a new point is computed by a function $g(\cdot, r)$ specified as one of `"min"`, `"max"` or `"prod"`. The argument `recent` allows the user to consider either all or only the r most recent events.

The sequence of commands

```
R> inf1 <- rinfec(npoints = 100, alpha = 0.1, beta = 0.6, gamma = 0.5,
+  maxrad = c(0.075, 0.5), t.region = c(0, 50), s.distr = "uniform",
+  t.distr = "uniform", h = "step", g = "min", recent = "all",
+  inhibition = TRUE)
R> animation(inf1$xyt, cex = 0.8, runtime = 10)
```

generates one realisation of an infectious/inhibitory process and displays the realisation over the spatial intensity estimate.

When the spatial distribution is specified as `"poisson"`, its intensity can be defined by a function $\lambda(x, y, t, ...)$ or by a matrix representing the values of $\lambda(x, y)$ assumed constant over time. The following example illustrates the case of an intensity defined by a matrix, here corresponding to a kernel estimate of the spatial intensity of the `fmd` data-set. The commands

```
R> data("fmd")
R> data("northcumbria")
R> h <- mse2d(as.points(fmd[, 1:2]), northcumbria, nsmse = 30, range = 3000)
R> h <- h$h[which.min(h$mse)]
R> Ls <- kernel2d(as.points(fmd[, 1:2]), northcumbria, h, nx = 50, ny = 50)
R> 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", g = "min",
+  recent = 1, lambda = Ls$z, inhibition = FALSE)
R> image(Ls$x, Ls$y, Ls$z, col = grey((1000:1) / 1000))
R> polygon(northcumbria, lwd = 2)
R> animation(inf2$xyt, add = TRUE, cex = 0.7, runtime = 15)
```

generate one realisation of an infectious/contagious process in a given space-time region and display the realisation. Figure 12 illustrate realisations of the infectious processes "`inf1`" (left) and "`inf2`" (right) defined above.

3.5. Log-Gaussian Cox processes

A Cox process is a doubly stochastic point process formed as an inhomogeneous Poisson process with a stochastic intensity. Such processes were introduced by Cox (1955) in one temporal dimension. Their definition in space and time is:

1. $\{\Lambda(s,t) : s \in S, t \in T\}$ is a non-negative-valued stochastic process.
2. Conditional on $\{\Lambda(s,t) = \lambda(s,t) : s \in S, t \in T\}$, the events form an inhomogeneous Poisson process with intensity $\lambda(s,t)$.

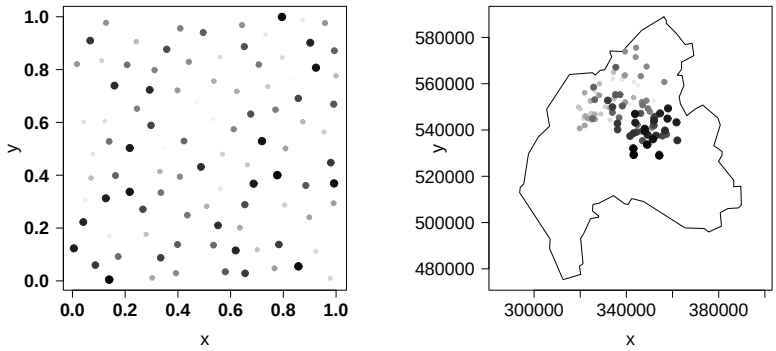


Figure 12: Realisations of the interaction processes "`inf1`" (left) and "`inf2`" (right) defined in examples.

Suppose that $Z = \{Z(s,t); s \in S, t \in T\}$ is a real-valued Gaussian process with mean $\mu(s,t) = \mathbb{E}[Z(s,t)]$ and covariance function $c\big((s_i,t_i),(s_j,t_j)\big) = \mathrm{Cov}\big(Z(s_i,t_i), Z(s_j,t_j)\big)$. If the intensity function is defined by $\Lambda(s,t) = \exp\{Z(s,t)\}$, then the corresponding process Y is a Log-Gaussian Cox process (Brix and Diggle 2001; Møller *et al.* 1998).

Simulation

The simulation of a log-Gaussian Cox process in the **stpp** package uses the same thinning algorithm as was used for an inhomogeneous Poisson process, but preceded by a simulation of the underlying Gaussian process. For a given covariance function $c\big((s_i,t_i),(s_j,t_j)\big)$ and a given mean $\mu(s,t)$ for the Gaussian process the algorithm is as follows.

1. *Generate a realisation of a Gaussian field, with covariance function* $c\big((s_i,t_i),(s_j,t_j)\big)$ *and mean* $\mu(s,t)$.
2. *Define* $\lambda(s,t) = \exp\{Z(s,t)\}$ *and an upper bound* λ_{max} *for* $\lambda(s,t)$.
3. *Simulate a homogeneous Poisson process with intensity* λ_{max}*; denote by* N *the number of points so generated.*
4. *Thin the simulated process as follows,*
   (a) *Compute* $p = \lambda(s,t)/\lambda_{max}$ *for each point* (s,t) *of the homogeneous Poisson process.*
   (b) *Generate a sample* u *of size* N *from the uniform distribution on* $(0,1)$.
   (c) *Retain the* $n \leq N$ *locations for which* $u \leq p$.

The **stpp** package implements an exact simulation of a stationary Gaussian random field on a $n_1 \times n_2 \times n_3$ grid, based on circulant embedding of the covariance function (Chan and Wood 1997). See e.g., Appendix E in Møller and Waagepetersen (2003) for the simulation of Gaussian random fields.

In **stpp**, we have implemented separable, stationary covariance functions of the form

$$c(h,t) = c_s(h)c_t(t), h \in S, t \in T, \tag{6}$$

where $c_s(h)$ and $c_t(t)$ are purely spatial and purely temporal covariance functions, respectively. All models are isotropic $c_s(h) = c(\|h\|)$ or $c_t(t) = c(|t|)$; they include the Matérn class, the Cauchy class and the wave class (see Table 1). Note that the Matérn covariance is defined in terms of the modified Bessel function K_ν.

Class	Functional form
Exponential	$c(r) = \sigma^2 \exp(-r),\ \sigma \geq 0$
Stable	$c(r) = \sigma^2 \exp(-r^\alpha),\ \ \alpha \in [0, 2],\ \sigma \geq 0$
Matérn	$c(r) = \sigma^2 \frac{(\alpha r)^\nu}{2^{\nu-1}\Gamma(\nu)} K_\nu(\alpha r),\ \nu > 0,\ \alpha > 0,\ \sigma \geq 0$
Cauchy	$c(r) = \sigma^2 (1 + r^2)^{-\alpha},\ \ \alpha > 0,\ \sigma \geq 0$
Wave	$c(r) = \sigma^2 \frac{\sin(r)}{r}$ if $r > 0,\ \ c(0) = 1,\ \sigma \geq 0$

Table 1: *Some classes of isotropic covariance functions.*

We have also implemented non-separable covariance functions of the form

$$c(h, t) = \psi(t)^{-\alpha_6} \phi\left(\frac{h}{\psi(t)}\right), \tag{7}$$

where $\phi(r), r \geq 0$ is a completely monotone function depending on parameters α_1 and α_2 and $\psi(r), r \geq 0$ is a positive function with a completely monotone derivative, depending on parameters α_1, α_2 and α_3. The model therefore depends on six parameters α_1 to α_6. The implemented choices for the function ϕ are:

- the stable model $\phi(r) = \exp(-r^{\alpha_1})$, if $\alpha_2 = 1$,
- the Cauchy model $\phi(r) = (1 + r^2)^{-\alpha_1}$, if $\alpha_2 = 2$.

The implemented choices for the function ψ are:

- $\psi^2(r) = (r^{\alpha_3} + 1)^{\alpha_4}$, if $\alpha_5 = 1$,
- $\psi^2(r) = (\alpha_4^{-1} r^{\alpha_3} + 1)/(r^{\alpha_3} + 1)$, if $\alpha_5 = 2$,
- $\psi^2(r) = -\log(r^{\alpha_3} + 1/\alpha_4)/\log \alpha_4$, if $\alpha_5 = 3$

The permissible ranges of the parameters of (7) are given in Table 2. See Gneiting, Genton, and Guttorp (2007) and references therein for details on these parameters.

Another non-separable covariance function is the de Cesare covariance function (De Cesare, Myers, and Posa 2001; De Iaco, Myers, and Posa 2002) defined by

$$c(h, t) = (1 + h^{\alpha_1} + t^{\alpha_2})^{\alpha_3},$$

where $\alpha_1, \alpha_2 \in [1, 2]$ and $\alpha_3 \geq 3/2$.

Examples

The function `rlgcp` generates realisations of a log-Gaussian Cox process. The covariance of the Gaussian process may be separable or not, as specified by the argument `separable`.

α_1	α_2	α_3	α_4	α_5	α_6
$[0, 2]$	1	$(0, 2]$	$(0, 1]$	1	$[2, \infty)$
$(0, \infty)$	2	$(0, 2]$	$(0, 1]$	1	$[2, \infty)$
$[0, 2]$	1	$(0, 2]$	$(0, 1]$	2	$[2, \infty)$
$(0, \infty)$	2	$(0, 2]$	$(0, 1]$	2	$[2, \infty)$
$[0, 2]$	1	$(0, 2]$	$(0, 1]$	3	$[2, \infty)$
$(0, \infty)$	2	$(0, 2]$	$(0, 1]$	3	$[2, \infty)$

Table 2: *Permissible ranges of the parameters defining the Gneiting model (7).*

The argument `model` is a vector of length 1 or 2 specifying the covariance model(s) for 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. When `separable = TRUE` and `model` is of length 1, the spatial and temporal covariances are assumed to belong to the same class of covariances, choices for which are `"matern"`, `"exponential"`, `"stable"`, `"cauchy"` and `"wave"`.

When `separable = FALSE`, `model` must be of length 1 and must be either `"gneiting"` or `"cesare"`. In all cases, parameters of the covariance models are specified by the vector argument `param`, whilst the mean and variance of the Gaussian process are specified through the arguments `mean.grf` and `var.grf`.

The thinning algorithm used to generate the space-time pattern depends on the space-time intensity $\Lambda(x, y, t)$, which is a evaluated on a `nx`×`ny`×`nt` grid. The larger the grid size, the slower are the simulations. Simulation time is also longer when the argument `exact` takes the value `TRUE`, providing an exact simulation rather than an approximation; see Chan and Wood (1999) for details about the exact and approximate procedures.

The sequence of commands

```
R> 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)
R> N <- lgcp1$Lambda[,,1]
R> for(j in 2:(dim(lgcp1$Lambda)[3])){N <- N + lgcp1$Lambda[, , j]}
R> image(N, col = grey((1000:1) / 1000)) ; box()
R> animation(lgcp1$xyt, cex = 0.8, runtime = 10, add = TRUE,
+  prevalent = "orange")

R> 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)
R> N <- lgcp2$Lambda[, , 1]
R> for(j in 2:(dim(lgcp2$Lambda)[3])){N <- N + lgcp2$Lambda[, , j]}
R> image(N, col = grey((1000:1) / 1000)) ; box()
R> animation(lgcp2$xyt, cex = 0.8, runtime = 10, add = TRUE,
+  prevalent = "orange")
```

generates a realisation of each of two log-Gaussian Cox processes, one with a non-seperable and one with a separable covariance structure, and displays the realisations superimposed on a grey-scale image of the spatial intensity.

Figure 13 illustrates spatial intensities (left) and realisations (right) of the log-Gaussian Cox processes "`lgcp1`" (top) and "`lgcp2`" (bottom) defined above.

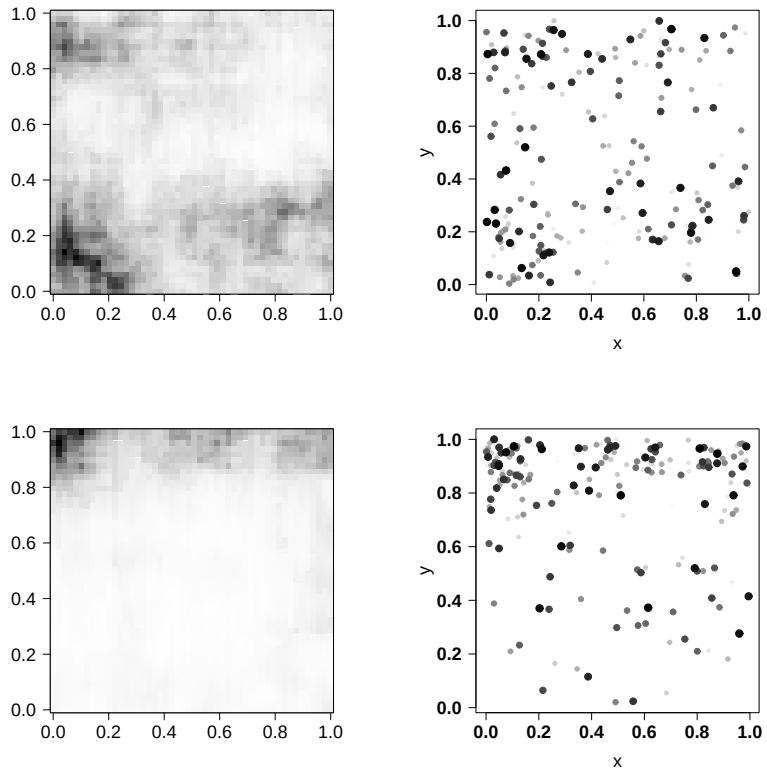


Figure 13: Spatial intensities (left) and realisations (right) of the log-Gaussian Cox processes "`lgcp1`" (top) and "`lgcp2`" (bottom) defined in examples.

References

Adler D, Murdoch D (2012). ***rgl**: 3D Visualization Device System (OpenGL)*. URL `http://rgl.neoscientists.org`.

Baddeley A, Møller J, Waagepetersen R (2000). "Non- and Semi-parametric Estimation of Interaction in Inhomogeneous Point Patterns." *Statistica Neerlandica*, **54**(3), 329–350.

Baddeley A, Turner R (2005). "**Spatstat**: An R Package for Analyzing Spatial Point Patterns." *Journal of Statistical Software*, **12**(6), 1–42.

Berman M, Diggle P (1989). "Estimating Weighted Integrals of the Second-Order Intensity of a Spatial Point Process." *Journal of the Royal Statistical Society B*, **51**, 81–92.

Bowman A, Gibson L, Scott M, Crawford E (2010). "Interactive Teaching Tools for Spatial Sampling." *Journal of Statistical Software*, **36**(13), 1–17.

Brix A, Diggle P (2001). "Spatiotemporal Prediction for Log-Gaussian Cox Processes." *Journal of the Royal Statistical Society B*, **63**(4), 823–841.

Chan G, Wood A (1997). "Algorithm AS 312: An Algorithm for Simulating Stationary Gaussian Random Fields." *Journal of the Royal Statistical Society C*, **46**, 171–181.

Chan G, Wood A (1999). "Simulation of Stationary Gaussian Vector Fields." *Statistics and Computing*, **9**, 265–268.

Cox D (1955). "Some Statistical Methods Connected with Series of Events." *Journal of the Royal Statistical Society B*, **17**(2), 129–164.

Cox D, Isham V (1980). *Point processes*. Chapman and Hall/CRC, London.

Cressie N (1993). *Statistics for Spatial Data*. Revised edition. John Wiley & Sons, New York.

Cressie N, Wikle C (2011). *Statistics for Spatio-Temporal Data*. 1st edition. John Wiley & Sons, New York.

Daley D, Vere-Jones D (2003). *An Introduction to the Theory of Point Processes. Vol. I*. Probability and its Applications, 2d edition. Springer-Verlag, New York.

De Cesare L, Myers D, Posa D (2001). "Estimating and Modeling Space-Time Correlation Structures." *Statistics and Probability Letters*, **51**(1), 9–14.

De Iaco S, Myers D, Posa D (2002). "Nonseparable Space-Time Covariance Models: Some Parametric Families." *Mathematical Geology*, **34**(1), 23–42.

Diggle P (2003). *Statistical Analysis of Spatial Point Patterns*. 2nd edition. Edward Arnold, London.

Diggle P (2006). "Spatio-Temporal Point Processes: Methods and Applications." In B Finkenstadt, L Held, V Isham (eds.), *Statistical Methods for Spatio-Temporal Systems*, Monographs on Statistics and Applied Probability, pp. 1–45. Chapman and Hall/CRC, London.

Diggle P, Gabriel E (2010). "Spatio-Temporal Point Processes." In A Gelfand, P Diggle, M Fuentes, G P (eds.), *Handbook of Spatial Statistics*, pp. 449–461. Chapman and Hall/CRC, London.

Diggle P, Kaimi I, Abellana R (2010). "Partial Likelihood Analysis of Spatio-Temporal Point Process Data." *Biometrics*, **66**, 347–354.

Gabriel E (2012). "Estimating Second-Order Characteristics of Inhomogeneous Spatio-Temporal Point Processes: Influence of Edge Correction Methods and Intensity Estimates." Submitted.

Gabriel E, Diggle P (2009). "Second-Order Analysis of Inhomogeneous Spatio-Temporal Point Process Data." *Statistica Neerlandica*, **63**(1), 43–51.

Gneiting T, Genton M, Guttorp P (2007). "Geostatistical Space-Time Models, Stationarity, Separability and Full Symmetry." In B Finkenstadt, L Held, V Isham (eds.), *Statistical Methods for Spatio-Temporal Systems*, pp. 151–175. Chapman and Hall/CRC, Boca Raton.

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

Keeling M, Woolhouse M, Shaw D, Matthews L, Chase-Topping M, Haydon D, Cornell S, Kappey J, Wilesmith J, Grenfell B (2001). "Dynamics of the 2001 UK Foot and Mouth Epidemic: Stochastic Dispersal in a Heterogeneous Landscape." *Science*, **294**, 813–817.

Møller J, Ghorbani M (2012). "Aspects of Second-Order Analysis of Structured Inhomogeneous Spatio-Temporal Point Processes." *Statistica Neerlandica*, **66**(4), 472–491.

Møller J, Syversveen A, Waagepetersen R (1998). "Log Gaussian Cox Processes." *Scandinavian Journal of Statistics*, **25**(3).

Møller J, Waagepetersen R (2003). *Statistical Inference and Simulation for Spatial Point Processes*. Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, London.

Neyman J, Scott E (1958). "Statistical Approach to Problems of Cosmology (with discussion)." *Journal of the Royal Statistical Society B*, **20**, 1–43.

Pebesma E (2004). "Multivariable Geostatistics in S: the **gstat** Package." *Computers & Geosciences*, **30**, 683–691.

Pebesma E, Graeler B, Gottfried T (2012). ***spacetime**: Classes and Methods for Spatio-Temporal Data*. URL http://cran.r-project.org/.

R Development Core Team (2012). "R: A Language and Environment for Statistical Computing." *R Foundation for Statistical Computing Vienna Austria*. URL http://www.r-project.org.

Rowlingson B, Diggle P (1993). "**Splancs**: Spatial Point Pattern Analysis Code in S-Plus." *Computers and Geosciences*, **19**(5), 627–655.

Silverman B (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, London.

Tanemura M (1979). "On Random Complete Packing by Discs." *Annals of the Institute of Statistical Mathematics*, **31**(Part B), 351–365.

Taylor B, Davies T, Rowlingson B, Diggle P (2012). ***lgcp**: Log-Gaussian Cox Process*. URL http://cran.r-project.org/web/packages/lgcp/vignettes/lgcp.pdf.

Wand M, Ripley B (2012). ***KernSmooth**: Functions for Kernel Smoothing for Wand and Jones (1995)*. URL http://cran.r-project.org/.

Zheng P, Diggle P (2012). ***spatialkernel**: Nonparameteric Estimation of Spatial Segregation in a Multivariate Point Process*. URL http://cran.r-project.org/.

Zhuang J, Ogata Y, Jones D (2002). "Stochastic Declustering of Space-Time Earthquake Occurrences." *Journal of the American Statistical Association*, **97**(458), 369–380.

Affiliation:

Edith Gabriel
Département de Mathématiques
Université d'Avignon et des Pays de Vaucluse
33 Rue Louis Pasteur
84000 Avignon, France
E-mail: edith.gabriel@univ-avignon.fr
URL: http://edith.gabriel.pagesperso-orange.fr/

Barry Rowlingson
Division of Medicine
Lancaster University
Lancaster LA1 4YB, UK
E-mail: b.rowlingson@lancaster.ac.uk

Peter Diggle
Division of Medicine
Lancaster University
Lancaster LA1 4YB, UK
E-mail: p.digglel@lancaster.ac.uk

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd