Non-Parametric Analysis of Spatial and Spatio-Temporal Point Patterns

Abstract:
The analysis of spatial and spatio-temporal point patterns is becoming increasingly necessary, given the rapid emergence of geographically and temporally indexed data in a wide range of fields. Non-parametric point pattern methods are a highly adaptable approach to answering questions about the real-world using complex data in the form of collections of points. Several methodological advances have been introduced in the last few years. This paper examines the current methodology, including the most recent developments in estimation and computation, and shows how various R packages can be combined to run a set of non-parametric point pattern analyses in a guided and intuitive way. An example of non-specific gastrointestinal disease reports in Hampshire, UK, from 2001 to 2003 is used to illustrate the methods, procedures and interpretations.

cite pdf tweet

AUTHORS
Jonatan A. González
Paula Moraga

AFFILIATIONS
King Abdullah University of Science and Technology (KAUST)
King Abdullah University of Science and Technology (KAUST)

PUBLISHED
Aug. 25, 2023

RECEIVED
Jan 9, 2022

DOI
10.32614/RJ-2023-025

VOLUME
15/1

PAGES
65 - 82

1 Introduction

A spatial point pattern $X = \{\mathbf{u}_i\}_{i=1}^n$ is a random collection of points observed within a bounded region of the plane, $W \subset \mathbb{R}^2$. Spatial point patterns arise in a broad range of applied disciplines such as climatology, ecology, epidemiology, and seismology. A spatial point pattern can represent the locations of forest fires (Turner 2009), the occurrence of species (Moraga 2021) or the residence locations of people with a certain disease (Moraga and Montes 2011). When the points evolve in space and time, we call them *spatio-temporal point patterns*. In this case, we treat the data as being generated as a snapshot in space-time, and they may be viewed as a spatial point process with a further (temporal) dimension (González et al. 2016). Similarly to the spatial case, we consider a spatio-temporal point pattern as a countable set of random points $\{(\mathbf{u}_i, v_i)\}_{i=1}^n$ (non-overlapping), where $\mathbf{u}_i$ represents the spatial location of the ith event and all of them belong to a bounded planar region $W \subset \mathbb{R}^2$. Here, v_i is associated with time and it belongs to a compact positive interval $T \subset \mathbb{R}_+$. $W \times T$ is usually called the spatio-temporal observation window. Examples of spatio-temporal point patterns include the locations of individuals with an infectious disease in which the time of infection is known (Diggle 2013), the starting (or ending) locations of tornadoes with their registered times (González et al. 2019) or the location and time at which some crime was committed (Rodrigues and Diggle 2012). Point pattern analyses may also be performed in smaller dimensions, such as straight segments (Daley and Vere-Jones 2003). For example, we can consider a time segment as a straight segment and record the exact time the heartbeat of a patient with arrhythmia occurs during an electrocardiogram. In addition, we can also have *marked point patterns* in which each of the points has some additional characteristics. For example, a marked point pattern may represent a pattern of beta-type ganglion cells in a cat's retina that can be classified anatomically as the marks "on" or "off" (Wässle et al. 1981).

When dealing with point patterns, it is desirable to learn as much as possible about the probability distribution that governs the occurrence of points in a given point pattern. This is a difficult task since, in general, there is only a single set of points or realisation of the point process. However, many methods have been developed that serve to estimate specific characteristics of the distribution, such as the intensity, which describes the expected number of points per unit area; or the clustering or aggregation behaviour, that is, whether the points have a particular attraction or repulsion mechanism. Statistical models for point processes can also be formulated, and these may consider previous knowledge to describe trends, variability, and different factors that may impact them. Point patterns are usually considered a random sample of a stochastic mechanism called *point process*. The most typical families of point processes are *Poisson processes*. There are two classes of Poisson processes; the *homogeneous*, whose realisations in any subset of the observation window are uniform random samples with a constant mean λ, and the *inhomogeneous*, obtained by replacing the intensity λ by a spatial, temporal or spatio-temporal varying one (Illian et al. 2008; Diggle 2013). Homogeneous Poisson processes are benchmark models for spatial and spatio-temporal point pattern data, often called *complete spatial random (CSR)* or *complete spatio-temporal random (CSTR)* point processes; i.e., point processes whereby point events appear in a completely random fashion.

A large body of literature has been developed in recent years to study point patterns and many statistical software packages have been developed for their estimation and analysis. For instance, packages such as spatial (Venables and Ripley 2002) or splancs (Rowlingson and Diggle 1993) initially developed for S-PLUS but available on CRAN are classics for displaying and analysing spatial point pattern data. Packages as sp (Pebesma and B. 2005; Bivand et al. 2013), or more recently, sf (Pebesma 2018), were created to provide a uniform interface for handling 2D and 3D data. spatstat (Baddeley and Turner 2005; Baddeley et al. 2015) is one of the most popular and improved packages for analysing spatial point patterns It is divided into several subpackages: spatstat.utils, spatstat.sparse, spatstat.data, spatstat.geom, spatstat.random, spatstat.explore, spatstat.model and spatstat.linnet. Some extra specialised packages such as smacpod or sparr (Davies et al. 2018) have methods for analysing case-control point patterns; or spatgraphs for graphs-related analyses of locations in 1D, 2D, 3D.

Even though authors have considered the combination of space and time in theory, and nowadays, spatial data often includes a time component, only a few examples show this in practice. This problem may be due to the complexity of the computations, added to the fact that the authors tend to delve into particular contexts. There are some tools available for dealing with spatio-temporal point pattern data. The stpp package (Gabriel et al. 2013) covers many models and functional descriptors related to point process methods to study spatio-temporal phenomena. The PtProcess package (Harte 2010) was designed to fit time-indexed point process models in the same simplified fashion as generalised linear models software. The lgcp (Taylor et al. 2013), as well as the inla (Rue et al. 2009; Moraga 2019); http://www.r-inla.org and inlabru (Bachl et al. 2019) packages, are suitable options for statistical inference in space and time.

In this paper, we demonstrate how R (R Core Team 2023) can be used to study spatial and spatio-temporal point process data using a dataset of geographic locations and times of individuals with non-specific gastrointestinal infections in Hampshire, UK, from 2001 to 2003 (Diggle et al. 2003). We consider the data as realisations of spatial and spatio-temporal point processes that lack spatial and temporal homogeneity; i.e., the expected number of points in each area unit of the study region depends on their location and time. Consequently, the first problem we solve is the intensity estimation. We then investigate the nature of the stochastic interactions between the process points after adjusting for spatial and spatio-temporal inhomogeneity. We use second-order descriptors to describe these interactions.

Along with this paper, we employ several R packages to successfully analyse the complex point patterns seen in both the spatial and the spatio-temporal settings. For dealing with spatial point patterns, we mainly use the spatstat package. In addition, we use the sparr package for analysing the relative risk and the GET package (Myllymäki and Mrkvička 2020) for testing hypotheses with envelopes. For the spatio-temporal analysis, we use the stpp package. We also make use of other R packages such as parallel, foreach (Microsoft and Weston 2022b) and doParallel (Microsoft and Weston 2022a) to optimise the computing time by dividing the burden by several processors. An R script containing all the necessary code for reading the data and running the analyses shown in this paper is provided as supplementary material.

Spatio-temporal point pattern data

We use the geographic locations and time information of individuals who had non-specific gastrointestinal infections in Hampshire, UK, between 2001 and 2003 (Diggle et al. 2003). These data come from the *Ascertainment and Enhancement of Gastrointestinal Infection Surveillance and Statistics* (AEGISS) project (Diggle et al. 2003), which, using the Hampshire area as a test, seeks to identify irregularities, i.e., high or low intensity spots in the spatio-temporal distribution of non-specific gastrointestinal infections in the UK.

The data consist of a collection of points $\{(\mathbf{u}_i, v_i)\}_{i=1}^n$, where $\mathbf{u}_i$ is a spatial coordinate with two components, i.e., $\mathbf{u}_i = (a_i, b_i), a_i, b_i \in \mathbb{R}$, and $v_i \in \mathbb{R}_+$ is a temporal coordinate. These points represent the locations and times of daily healthcare providers reports of non-specific gastrointestinal disease in Hampshire from 2001 to 2003. After cleaning the data to remove non-Hampshire residents (out of the Hampshire region) and multiple (coincident) points, there were $n = 10443$ cases of non-specific gastrointestinal disease by a phone-in triage service operating within the UK's National Health Service. The command sequence

```
library(spatstat)
library(viridis)
load("AegissData.RData")
```

loads the data in the local environment of R. Each location corresponds to the centroid of the unit post-code of the residential address of the caller. The unit of distance is one kilometre. The unit of time is one day, with day 1 corresponding to 1 January 2001. We can inspect the first spatio-temporal coordinates of our dataset by writing

```
head(as.data.frame(Aegiss))
```

```
       x      y marks
1 478.55 134.85     1
2 458.15 107.35     1
3 449.05 128.95     1
4 450.75 106.95     1
5 461.05  99.55     1
6 446.35 109.15     1
```

Notice that x and y are the coordinates. Times are not labelled as dates, but they are integers, and their column is named as 'marks'. Finally, the planar region that encloses the points corresponds to a 120-sided polygon representing the boundary of the study region of Hampshire. To visualise the spatio-temporal point pattern, we can depict a spatial map using the `symbolmap()` function of the spatstat.geom package setting the times as colours. The colours are selected by using the viridis (Garnier et al. 2023) (Figure 1).

```
Times <- Aegiss$marks
timelabels <- as.Date(Times - 1, origin = "2001-01-01")
colmap <- viridis(length(Times))
sy <- symbolmap(pch = 21, col = "black", bg = colmap, range = range(timelabels))
```

We use the locations and times of the point pattern and the population density of Hampshire to study the point process that generates these data. We obtain the mean population density of Hampshire from WorldPop (University of Southampton). The population information detailed for each squared kilometre is available in a yearly basis. Figure 1 shows the point pattern of the cases and the population density. First, we save only the spatial locations with no marks by writing `X <- unmark(Aegiss)`. Then, we attach the date labels to the points using the operator `%mark%`. Note that `symap = sy` sets the colour and point shapes previously defined and that the six dates in the legend are a sample to represent the colour variation. To summarise the population density, given in a set of three images, 2001, 2002 and 2003, we use the function `im.apply()` of the package spatstat.geom to average them.

```
par(mfrow = c(1, 2), mar = c(0,0,2,0.5))
X <- unmark(Aegiss)
PopDens <- im.apply(Population, mean)
plot(X %mark% timelabels, symap = sy,
     main = "Gastrointestinal disease reports")
plot(PopDens, col = viridis(prod(PopDens$dim)), log = T, box = F,
     main = "Population intensity")
```

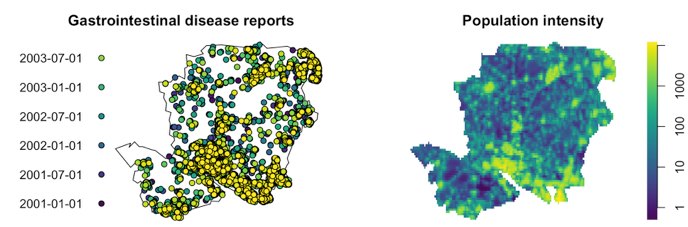

Figure 1: Left: Locations of 10443 reports of non-specific gastrointestinal disease in Hampshire from January 2001 to July 2003. The time is treated as a quantitative mark where darker dots correspond to the older events, and lighter dots match the most recent reports. Right: Hampshire population density averaged over the years 2001-2003. The colour map is evenly-spaced on a logarithmic scale.

2 Spatial analysis

Spatial intensity and relative risk

The *first-order intensity function* or the *intensity* of a point process can be defined as the expected number of events per unit space, or mathematically (Diggle 2013),

$$\lambda(\mathbf{u}) = \lim_{|\mathrm{d}\mathbf{u}| \to 0} \frac{\mathbb{E}N(\mathrm{d}\mathbf{u})}{|\mathrm{d}\mathbf{u}|},$$

where $N()$ denotes the number of points of a region. In the inhomogeneous case, we can compute a kernel smoothed intensity function from a point pattern $\{\mathbf{u}_i\}_{i=1}^n$ without the temporal coordinates. The estimator is given by

$$\hat{\lambda}(\mathbf{u}) = \frac{1}{e^2(\mathbf{u})} \sum_{i=1}^{n} \kappa_\epsilon^2(\mathbf{u} - \mathbf{u}_i), \quad \text{and} \quad e^2(\mathbf{u}) = \int_W \kappa_\epsilon^2(\mathbf{s} - \mathbf{u}) \mathrm{d}\mathbf{s}, \tag{1}$$

where $\mathbf{u} \in W$ and κ_ϵ^2 is a two-dimensional Gaussian kernel. $e()$ is known as *uniform edge correction* and is intended to correct the bias of the estimation at the edges of the region (Baddeley et al. 2015). One of the most challenging considerations is selecting the bandwidth ϵ; there is no single best procedure for doing that, but the estimation can be improved by using a spatially varying bandwidth despite its computational complexity (Davies and Baddeley 2018). For computing the adaptive estimate of the intensity function, we use the `adaptative.density()` function of the spatstat.explore package,

```
# This takes roughly 12 seconds to be executed in an
# iMAC 2019, 3GHz intel core i5 processor with 16GB of RAM (hereinafter)
SpatialDens <- adaptive.density(X, method = "kernel", edge = T)
```

The estimated intensity is shown in Figure 2 (left). We can see a high concentration of cases (more than two per square kilometre) in Southampton and its coastal environs. More to the north, there are also local foci in Winchester (the centre of the study region) and the primary urban centres located in the north and northeast of the study region.

The *kernel log relative risk* or *density ratio function* is a descriptor aimed to compare two estimated densities on the study region W. In epidemiology, it is handy to examine disease risk fluctuations based on case and control samples. In recent years, many improvements have been developed to the estimation methodology, including spatially adaptive smoothers and inference based on asymptotic theory (Davies et al. 2018). The related techniques are implemented in the sparr package. Given two spatial point patterns, X, with intensity $\hat{\lambda}(\mathbf{u}|X)$ (that can be thought of as cases) and Y, with intensity $\hat{\lambda}(\mathbf{u}|Y)$ (that can be thought of as controls); the estimated log relative risk is defined as

$$\hat{r}(\mathbf{u}) = \log\left\{\frac{\hat{\lambda}(\mathbf{u}|X)}{\hat{\lambda}(\mathbf{u}|Y)} \cdot \frac{n_Y}{n_X}\right\}, \quad \mathbf{u} \in W, \tag{2}$$

where n_X and n_Y are the number of points of X and Y, respectively. When $\hat{r}(\mathbf{u}) \approx 0$, the densities are roughly the same; peaks greater than zero mean a higher local ratio of cases to controls, and $\hat{r}(\mathbf{u}) < 0$ indicates lack of concentration of cases than controls.

Here, we compute the relative risk by comparing the point pattern of gastrointestinal infections and a point pattern with controls derived from a random sample of the underlying population. We first generate a set of random controls by simulating $n = 1443$ locations with intensity proportional to the population density using the `rpoispp()` function of package spatstat.random. In order to do that, we calculate the population density as the population intensity normalised by its mass (its integral over the observation window) calculated by using the function `integral.im()` of the package spatstat.geom in each region of Hampshire. Then we can evaluate expressions involving one or more pixel images through the `eval.im()` function of package spatstat.geom,

```
N <- integral.im(PopDens)
n <- X$n
ControlDens <- eval.im((PopDens / N) * n)
Controls <- rpoispp(ControlDens)
```

Then we assign Hampshire's window to the new point pattern by using the `Window()` function of package spatstat.geom,

```
Window(Controls) <- X$window
```

For estimating the relative risk, we use the sparr package to perform an adaptive smoothing adequately designed to deal with the reduced smoothing in densely populated areas. The code

```
# This takes roughly 22 seconds to be executed
library(sparr)
RelativeRisk <- risk(f = X, g = Controls, adapt = T,
                     pilot.symmetry = "pooled", tolerate = T)
```

takes both point patterns (cases `f` and controls `g`), computes an adaptive smoothing for estimating the risk, and computes asymptotic tolerance contours as per Hazelton and Davies (2009). Maps displayed in Figure 2 showing the intensity and relative risk estimates are produced by executing

```
par(mfrow = c(1, 2), mar = c(0,0,2,0.5))
plot(SpatialDens, col = viridis(1200), auto.axes = F,
     main = "Adaptative kernel intensity estimate", box = F)
plot(RelativeRisk, auto.axes = F, tol.type = "upper",
     main = "Relative risk estimate")
```

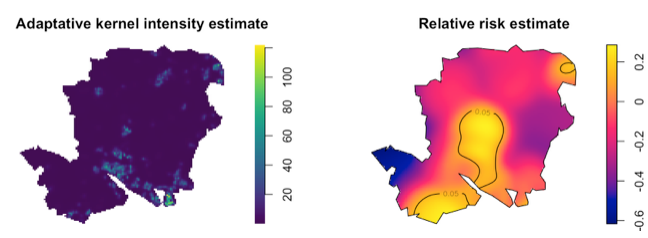

Figure 2: Left: Adaptive bandwidth kernel estimate of intensity for the gastrointestinal data. The intensity is expressed as number of reports per squared kilometre. Right: Adaptive spatial log-relative risk of non-specific gastrointestinal disease, with asymptotic tolerance contours for elevated risk.

Figure 2 (right) provides pointwise areas of significant high relative risk. The null hypothesis is $H_0 : r(\mathbf{u}) = 0$, against an alternative of increased local risk $H_1 : r(\mathbf{u}) > 0$ for a point $\mathbf{u} \in W$. We superimpose the contours corresponding to significant values on the log relative risk surface. They are known as tolerance contours and allow us to identify areas of high risk (Kelsall and Diggle 1995). The relative abundance of gastrointestinal disease reports appears to be significantly higher in some areas where the intensity is not necessarily higher. Nor does the relative risk appear to increase in Southampton (located on the southern coast), the region's most urbanised area.

Second-order descriptors

Pair correlation and K-functions

The *second-order intensity function* $(\lambda^{(2)}(\mathbf{u}_1, \mathbf{u}_2), \mathbf{u}_1, \mathbf{u}_2 \in W)$ is, in point processes, analogous to the covariance function in classical statistics (Baddeley et al. 2015). Also known as *product density function,* the second-order intensity function represents the infinitesimal two-dimensional joint distribution of the points. A more friendly summary statistic is the *pair correlation function*, which is the standardised probability density that an event occurs in each of two small spatial volumes $\mathrm{d}\mathbf{u}_1$ and $\mathrm{d}\mathbf{u}_2$. For CSR (Complete Spatial Random) point processes, the covariance is zero, and the pair correlation function takes the value of one (Diggle 2013). Different values than these benchmarks indicate how likely a pair of events will occur at the specified locations than in a CSR process with the same intensity; i.e., larger values indicate clustering, and smaller values indicate regularity. The pair correlation is defined as

$$g(\mathbf{u}_1, \mathbf{u}_2) = \frac{\lambda^{(2)}(\mathbf{u}_1, \mathbf{u}_2)}{\lambda(\mathbf{u}_1)\lambda(\mathbf{u}_2)}, \qquad \mathbf{u}_1, \mathbf{u}_2 \in W. \tag{3}$$

Whenever the intensity function of a point process is bounded away from zero and its pair correlation function depends only on the difference vector $r = ||\mathbf{u}_1 - \mathbf{u}_2||$, the point process is called *second-order intensity reweighted stationary* (SOIRS) and *isotropic*. For estimating the pair correlation function, consider a point pattern $X = \{\mathbf{u}_i\}_{i=1}^n$ with n points, then

$$\hat{g}(r) = \frac{1}{2\pi} \sum_{i=1}^{n} \sum_{j \neq i} \frac{\kappa_\epsilon(||\mathbf{u}_i - \mathbf{u}_j|| - r)}{\hat{\lambda}(\mathbf{u}_i)\hat{\lambda}(\mathbf{u}_j)||\mathbf{u}_i - \mathbf{u}_j||w_{ij}}. \tag{4}$$

Several parameters must be set before the estimation: a suitable interval for the spatial distances r, a bandwidth ϵ, a one-dimensional kernel $\kappa_\epsilon(\cdot)$ (Epanechnikov's kernel is usually recommended), a type of divisor (factor $||\mathbf{u}_i - \mathbf{u}_j||$ in the denominator of Eq. (4) can be replaced by r, this option brings a pole, i.e., an indetermination, in $r = 0$ (Wong and Stoyan 2021)); and an edge correction w_{ij} correcting for the loss of information of outsider points close to the edges. spatstat usually has smart defaults for these parameters based on the knowledge of the descriptors, which many authors have contributed. For this part, we use a fixed-bandwidth kernel estimate, with bandwidth set by Scott's rule of thumb (Scott 2015), through the function `bw.scott()` of the package spatstat.explore, as we need simulations where the intensity has to be estimated time after time. We employ the function `density.ppp()` of package spatstat.explore that provides a fixed bandwidth kernel smoothed intensity function from a point pattern according to Eq. (1).

```
sigmaD <- bw.scott(X)
MD <- density.ppp(X, diggle = T, positive = T, sigma = sigmaD)
MP <- density.ppp(X, diggle = T, sigma = sigmaD, at = "points")
```

In the above implementation, `diggle = T` sets the Jones-Diggle edge correction (Diggle 1985), `at = 'points'` means that the intensity is computed only at the points of X, and `sigma` is the bandwidth, which in this case is 3.5km for the horizontal axis and 4.19km for the vertical axis. In the case of the bandwidth for the pair correlation function, we employ a composite likelihood approach (Jalilian and Waagepetersen 2018) by using the function `bw.pcf()` of library spatstat.explore with the parameter `cv.method = 'compLik'`; the parameter `divisor = 'd'` refers to the term $||\mathbf{u}_i - \mathbf{u}_j||$ in Eq. (4), which can be replaced by r (Jalilian and Waagepetersen 2018), and `lambda` is the estimate of the intensity. We then write

```
# Time consuming: This takes roughly 24 minutes to be executed
bwG <- bw.pcf(X, cv.method = "compLik", divisor = "d", lambda = MP)
```

and obtain a bandwidth value of 2.7km. Note that this function can warn of undetermined contributions to the pair correlation. These caveats come from the divisor, which, when too small, can conflict with the numerical tolerance of R and eventually lead to indeterminacy. The spatstat package fixes this by simply removing the problematic values.

Before going further with the estimation of the pair correlation function, let us take a look into another classical second-order descriptor, the *inhomogeneous K-function*, which in the SOIRS case, can be defined as

$$K(r) = 2\pi \int_0^r s g(s) \mathrm{d}s.$$

For spatial Poisson point processes $K(r) = \pi r^2$. spatstat chooses a suitable range for r values based on some rules of thumb depending on geometric attributes of the process or the average intensity in the region. We decide to choose a proportion of 70% of the maximum value for r proposed by spatstat.explore through the `rmax.rule()` function, and set of 70 (71 including zero) different values for r,

```
r0 <- 0.7 * rmax.rule("K", X$window, intensity(X))
rr <- seq(0, r0, length.out = 71)
```

The classical estimator of the spatial K-function is given by

$$\hat{K}(r) = \sum_{i=1}^{n} \sum_{j \neq i} \frac{\mathbf{1}\left[||\mathbf{u}_i - \mathbf{u}_j|| \leq r\right]}{\hat{\lambda}(\mathbf{u}_i)\hat{\lambda}(\mathbf{u}_j)w_{ij}}. \tag{5}$$

Where $\mathbf{1}[\cdot]$ is the indicator function. An alternative descriptor defined as $L(r) - r = \sqrt{K(r)/\pi} - r$, and its straightforward estimator $\hat{L}(r) - r$ are intended for stabilising the variance (Besag 1977).

In addition to the estimation of the second-order descriptors, their standard deviation and confidence intervals are desirable for the analysis; this is not always simple, not even in the Poisson case, as there is dependence between the contributions of different points to the descriptor. However, we can use Monte Carlo simulation for computing envelopes through computational efforts. We use command `envelope()` of the spatstat.explore for getting the observed descriptor as well as its simulated counterparts.

We note that although Monte Carlo testing is generally used in point process contexts, it becomes invalid when the null hypothesis is composite, i.e., when parameters, such as the intensity function, must be estimated prior to the computing of the envelopes, which is our case for inference about the second-order descriptors. For dealing with such composite hypotheses, an extra set of simulations is needed to perform a balanced two-stage test (Baddeley et al. 2017). The GET package performs global envelope tests (where the significance level is controlled simultaneously for all the functions) for the second-order descriptors and provides a graphical interpretation (see, e.g., Myllymäki et al. 2017). We perform a global rank envelope test with the extreme rank length measure and retrieve the p-value. We warn the reader that this procedure of computing envelopes and two-stage Monte Carlo testing is computationally demanding. We calculate the first set of 39 envelopes for the centred L-function. This selection gives 39 (simulated) + 1 (observed) independent samples; it comes from classical settings of point process simulation (Ripley 1977; Baddeley et al. 2015, 2017). We use the `envelope.ppp()` function of the spatstat.explore package; this function computes simulation envelopes of a summary function,

```
# Time consuming: This takes roughly 3 minutes to be executed
L1 <- envelope(X, nsim = 39, savefuns = TRUE, fun = "Linhom", diggle = T,
               transform = expression(.-r), sigma = sigmaD, r = rr,
               simulate = expression(rpoispp(lambda = MD)), verbose = F)
```

Then we generate another set of inhomogeneous Poisson point patterns with the estimated intensity,

```
Simpatterns <- rpoispp(lambda = MD, nsim = 39)
```

We design a function to calculate the intensity for each point pattern and to compute envelopes with point patterns simulated from the new intensity,

```
simL <- function(rep) {
  sim_fit <- density.ppp(Simpatterns[[rep]], diggle = T,
                         sigma = sigmaD, positive = T)
  envelope(Simpatterns[[rep]], nsim = 39, savefuns = T, fun = "Linhom",
           transform = expression(.-r), r = rr, diggle = T, sigma = sigmaD,
           simulate = expression(rpoispp(lambda = sim_fit)))
}
```

Now we apply the function for every point pattern of the list to get a list of envelope objects. This process is very slow as 39 L-functions have to be computed for each point pattern. We employ the package foreach to use multiple processors to accelerate this procedure in combination with the parallel and doParallel packages. The function `detectCores()` of the package parallel automatically detects the computer's CPU cores and the function `registerDoParallel()` of the doParallel package is used to register a parallel backend in the computer. The `foreach() %dopar%` function returns a list with the result of applying a function over every element of the main argument by using the registered cores of the computer processor,

```
# Time consuming: This takes roughly 21 minutes to be executed
library(foreach)
library(doParallel)
c0 <- parallel::makeCluster(detectCores() - 1)
doParallel::registerDoParallel(c0)
L.ls <- foreach(i = 1:39, .packages = c("spatstat")) %dopar% {simL(i)}
parallel::stopCluster(c0)
```

Finally, we perform the adjusted test using the function `GET.composite()` of the GET package and plot the envelopes. In this case, we are interested in establishing a possible positive departure of the observed centred L-function and its simulated counterparts. Thus, we set `alternative = "greater"`, `type = 'erl'`, which performs a rank envelope test based on extreme rank lengths (Myllymäki et al. 2017) based on several minimal pointwise ranks; and use the ggplot2 (Wickham 2016) package for the plots,

```
library(GET)
library(ggplot2)
resL <- GET.composite(X = L1, X.ls = L.ls, type = "erl",
                      alternative = "greater", savefuns = T)
```

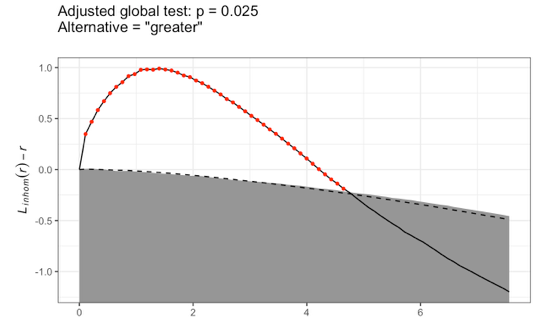

Figure 3: Graphical clustering tests of Hampshire data based on the L-functions. The gray bands represent the 95% global envelope. Red dots stand for the second-order descriptors outside the envelopes.

Figure 3 shows the output of the rank envelope test. The rejection is explained by the attractive behaviour of gastrointestinal disease cases at short distances. The red-dot line indicates this clustering behaviour up to 5km, which shows a departure from the narrow grey band that represents the 95% adjusted rank envelope, i.e., the rank envelope that considers the composite hypothesis. Note that the more simulations, the better results for the test (Myllymäki et al. 2017). The dashed black line represents the central function (the mean) of only the first set of simulations; note that this function is not the mean value of the grey ribbon as it comes from envelopes under the composite null hypothesis. We find a significant positive departure from the envelopes up to approximately 5km; this means that observed data are more clustered in distances below 5km than might be expected in Poisson point patterns with the same intensity, and this clustering is statistically significant.

3 Spatio-temporal analysis

Spatio-temporal intensity and relative risk

First-order separability

Spatio-temporal point processes analyses can be carried out when every data point has associated geographic and time information. In these cases, estimating the first-order intensity function constitutes a primary interest. The intensity function provides information about the joint distribution of the spatial and temporal locations. Usually, this distribution is simplified by assuming, for example, separability. A point process is referred to as when its spatio-temporal intensity function can be factorised as

$$\lambda(\mathbf{u}, v) = \lambda_1(\mathbf{u})\lambda_2(v), \quad \mathbf{u} \in W, v \in T,$$

where $\lambda_1(\cdot)$ and $\lambda_2(\cdot)$ are non-negative functions of space and time, respectively. Authors often take separability for granted as it is a very pragmatic working assumption, but it should be tested before choosing an estimator. There are some specialised separability tests available in the literature. We use the package kernstadapt (González and Moraga 2022b) to test separability in our example by considering a division of the temporal interval $T = [1, 1095]$ into 16 disjoint equally-spaced sub-intervals. We split the spatial window W into 20 disjoint rectangular subsets with roughly the same area. As the separability of the intensity might be thought of as the independence of two random variables, it can be tested by a simple χ^2-test as proposed by Ghorbani et al. (2021), where the null hypothesis is separability. The test bases on the counts of points in disjoint spatial and temporal sub-regions; when there are expected counts of individual cells below 5, the authors recommend a Fisher's test. We employ Fisher's test calculating the p-value by Monte Carlo simulation (5000 replicates):

(5000 replicates).

```
# This takes roughly 20 seconds to be executed
SepTest <- separability.test(Aegiss, nx = 5, ny = 4, nt = 16, nperm = 50000)
```

```
SepTest
```

```
    Separability test based on Fisher's for counting data

data:  Point pattern Aegiss
p-value = 0.00344
alternative hypothesis: The point pattern Aegiss is not spatio-temporal separable
```

As the p-value is significant, we opt for estimating the spatio-temporal intensity in a non-separable fashion. We estimate the intensity by employing the following estimator, which uses a Gaussian kernel for all coordinates (Choi and Hall 1999; González et al. 2019; Ghorbani et al. 2021)

$$\hat{\lambda}(\mathbf{u}, v) = \sum_{i=1}^{n} \frac{\kappa_\epsilon^2(\mathbf{u} - \mathbf{u}_i)\kappa_\delta^1(v - v_i)}{e^2(\mathbf{u}_i)e^1(v_i)}, \tag{6}$$

where

$$e^2(\mathbf{u}_i) = \int_W \kappa_\epsilon^2(\mathbf{u}_i - \mathbf{s})\mathrm{d}\mathbf{s}, \quad \text{and} \quad e^1(v_i) = \int_T \kappa_\delta^1(v_i - w)\mathrm{d}w;$$

and where κ_ϵ^2 and κ_δ^1 are the two- and one-dimensional Gaussian kernels with bandwidths ϵ and δ.

In this case, we use a fixed spatial bandwidth calculated by a rule of thumb that depends on the short size of the window and a temporal one calculated through the function `bw.nrd()`. Notice that the spatio-temporal estimator given in Eq.(6) resembles the spatial one given in Eq.(1), with an extra kernel weight for the temporal component. On the other hand, as the spatial edge correction, due to Diggle (1985), is already implemented in the function `density.ppp()` of spatstat.explore, we must calculate the temporal one. We employ the generic function `bw.nrd0()` of package stats for the temporal bandwidth, which uses Silverman's rule-of-thumb (Silverman 1986) for choosing the bandwidth of a Gaussian kernel density estimator. For computing the edge correction, we take advantage of the properties of the normal distribution noticing that $e^1(v_i) = \mathbb{P}(Y_i \in T)$ where $Y_i \sim N(v_i, \delta^2)$, indeed,

```
Times <- Aegiss$marks
bwt <- bw.nrd0(Times)
edgewt.t <- pnorm((max(Times) - Times) / bwt) - pnorm((min(Times)- Times) / bwt)
```

Then we proceed to calculate the non-separable estimator. To quicken the computation, for each fixed time t_0, we take temporal bins $t \in [t_0 - 3\delta, t_0 + 3\delta]$, as the normal distribution practically vanishes at larger or smaller values, then we compute the temporal kernel values and attach them to the spatial kernel as weights,

```
# This takes roughly 44 seconds to be executed
nonseparable <- function(time){
  contrib <- (Times >= time - 3 * bwt) & (Times <= time + 3 * bwt)
  Wh <- dnorm(x = Times[contrib], mean = time, sd = bwt) / edgewt.t[contrib]
  density.ppp(X[contrib], weights = Wh, diggle = TRUE)
}
nonsep <- lapply(unique(Times), nonseparable)
```

We have 1047 intensity estimates corresponding to each of the 1047 different times; note that 1047 is the size of the temporal coordinates set and not the maximum temporal value (1095). We decide to plot a subset of them, namely the estimates corresponding to times 01/01/2001, 01/01/2002, 01/01/2003 and 12/31/2003.

```
n.slices <- which(unique(timelabels) %in% as.Date(c("2001-01-01", "2002-01-01",
                                                    "2003-01-01", "2003-12-31")))
Snap <- list(nonsep[[n.slices[1]]], nonsep[[n.slices[2]]],
             nonsep[[n.slices[3]]], nonsep[[n.slices[4]]])
plot.imlist(Snap, equal.ribbon = T, ncols = 4, box = F, main = "", log = F,
            main.panel = unique(timelabels)[n.slices], col = viridis(1200),
            ribscale = 100,  mar.panel=c(0, 0, 1, 1), panel.end = X$window)
```

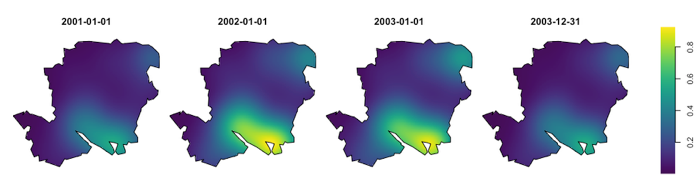

Figure 4: Temporal snapshots of the intensity $\hat{\lambda}(\mathbf{u}, v)$, of non-specific gastrointestinal reports in Hampshire, estimated in a non-separable way through Gaussian kernels. The units are rescaled by 100.

In the spatio-temporal case (Figure 4), we can see a behaviour in the intensity of cases much lower than in the spatial case (Figure 2 (left)) due to the temporal granularity. This scale difference is due to the number of events per unit of time, which is not considered in the only spatial estimation. When considering the temporal dimension, the total points must be divided by the number of bins in the temporal grid where the estimation is made. Therefore, there are fewer points per unit area at each time than only when spatial coordinates are considered.

The intensity maintains some urban centres as foci over time, especially those in the south, for example, in Portsmouth, where there is a high concentration of cases between 2002 and 2003.

In the next section, we show how to compute spatio-temporal descriptors. To do that, we need to know the intensity values at every point of the point pattern. We can extract the intensity values by using the `safelookup()` of spatstat.geom. Then, we attach those values, and the dates, as a mark to each point by using the operator `%mark%`,

```
lambda <- NULL
for (i in 1:length(nonsep)) {
  lambda <- c(lambda, safelookup(nonsep[[i]],
  Aegiss[marks(Aegiss) == unique(Times)[i]]))
}
PP <- X %mark% data.frame(time = Aegiss$marks, Lambda = lambda)
```

Spatio-temporal relative risk

The concept of relative risk straightforwardly extends to the spatio-temporal domain. In this regard, we must make additional considerations; for example, the spatial relative risk can be conditioned to a specific time (Davies et al. 2018). Analogous to the classic probability, in the spatio-temporal context, the idea of conditioning on a specific time $v = t_0$ comes naturally. This means that for a fixed instant t_0, we consider the spatial density function given that time. Mathematically, it is calculated by normalising the spatio-temporal density by the density of the temporal coordinates evaluated at the fixed time t_0. On the other hand, if it is not conditioned to a particular time, the risk function is a function of three variables (two spatial coordinates and one temporal), which can be drawn through snapshots. Thus, we have two possible risk functions, $r(\mathbf{u}, v)$ (if plotted at a fixed time, it represents an unnormalised slice of the three-dimensional function) and $r(\mathbf{u}|v = t_0)$ (a spatial-only snapshot of relative risk).

Given that we have population information for 2001, 2002 and 2003, we generate controls from these three population densities considering uniform times within each year. Then, we superimpose the three control point patterns into one using the `superimpose()` function of the spatstat.geom package. We obtain a spatio-temporal point pattern of controls that will serve as input to calculate the denominator of the relative risk.

```
Ni <- lapply(Population, integral.im)
ni <- lapply(split(Aegiss, cut(Aegiss$marks, breaks = c(0, 365, 730, 1095))),
             npoints)
ControlDensi <- mapply(function(D, N, n) {eval.im((D / N) * n)},
                       D = Population, N = Ni, n = ni, SIMPLIFY = F)
Controlsi <- mapply(function(L, Breaks){
  PP <- rpoispp(lambda = L)
  marks(PP) <- sample(x = (Breaks[1] + 1):Breaks[2],
                      size = npoints(PP), replace = T)
  return(PP)},
  L = ControlDensi, Breaks = list(c(0, 365), c(365, 730), c(730, 1095)),
  SIMPLIFY = F)

Controlsi <- superimpose(as.solist(Controlsi))
Controlsi <- unmark(Controlsi) %mark% Controlsi$marks$origMarks
Window(Controlsi) <- Window(Aegiss)
```

For the spatio-temporal risk estimation exercise, we return to the sparr package. We follow the recommendations of (Davies et al. 2018) and set *oversmoothing bandwidths* (a rule based on an optimal bandwidth upper bound for the asymptotic mean integrated squared error of the density estimate, this may be done using the command `OS.spattemp()`). The spatio-temporal densities can analysed through the command `spattemp.density()` of the sparr package.

```
# This takes roughly 4.87 minutes to be executed
Br <- OS.spattemp(Aegiss)
f.num <- spattemp.density(Aegiss, h = Br[1], lambda = Br[2])
d.den <- spattemp.density(Controlsi, h = Br[1], lambda = Br[2])
```

In this case, interest may rise about the anomalous regions of spatio-temporal risk considering joint and conditional risks (given a time t); i.e., we are interested in testing the null hypotheses $H_0 : r(\mathbf{u}, v) = 0$ or $H_0 : r(\mathbf{u}, |v = t) = 0$ against the alternatives of high risks. We employ the `spattemp.risk()` function of sparr to obtain a spatiotemporal relative risk surface based on the ratio of the two kernel estimates. In addition, since doing Monte Carlo simulations to obtain p-values is very complicated from the computational point of view, asymptotic approximations are implemented in sparr by setting `tolerate = T`,

```
# This takes roughly 5.47 minutes to be executed
st.risk <- spattemp.risk(f = f.num, g = d.den, tolerate = T)
```

We proceed to extract the unconditional and conditional risk slices,

```
risk.slices <- spattemp.slice(st.risk, tt = n.slices)
```

The variable `risk.slices` contains a list of lists of images, each of which corresponds to the requested times in n.slices and has the unconditional and conditional risk estimates and the p-values surfaces. We can plot the image estimates as follows:

```
plot.imlist(c(risk.slices$rr, risk.slices$rr.cond), equal.ribbon = T, ncols = 4,
            box = F, main = "", main.panel = c(timelabels[n.slices], rep("", 4)),
            mar.panel = c(0, 0, 1, 1), col = rocket(1200), ribmar = c(1, 3, 1, 2),
            ribwid = 0.4, panel.end = function(i,...){
                contour(c(risk.slices$P, risk.slices$P.cond)[[i]],
                        levels = 0.05, drawlabels = TRUE, ...)
                plot(X$window, add = T)})
```

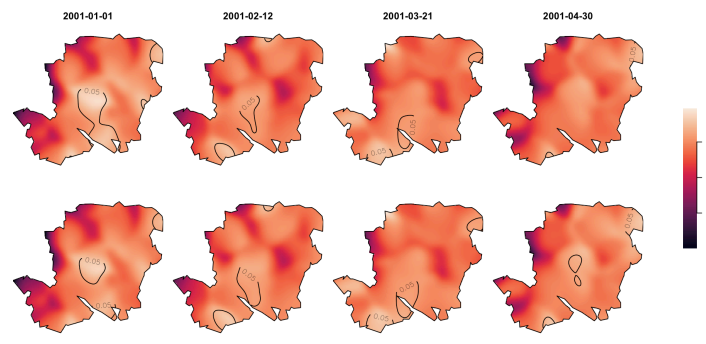

Figure 5: Temporal snapshots of the unconditional risk $\hat{r}(\mathbf{u}, v)$ (top) and conditional risk $\hat{r}(\mathbf{u}|v = t)$ (where t is $01/01/2001$, $01/01/2002$, $01/01/2003$ and $12/31/2003$) (bottom) of non-specific gastrointestinal disease in Hampshire. Solid lines delineate a statistical significant elevated risk at the 5% level.

In Figure 5, we can see how the high significant risk remains in the central Hampshire region. Although the two panels (top and bottom) look similar, there are differences. For example, in February 2001, we see how the conditional risk shows a larger statistically high area than its unconditional counterpart. We also see that the greatest risk is not present in the most densely populated areas (in this case, the south-central part of the region, where the capital is located).

Second-order descriptors

Pair correlation and K-functions

Let $\xi_1 = (\mathbf{u}_1, v_1), \xi_2 = (\mathbf{u}_2, v_2)$ be two points of $W \times T$. The second-order intensity function $(\lambda^{(2)}(\xi_1, \xi_2), \xi_1, \xi_2 \in W \times T)$ and the pair correlation function are defined in the spatio-temporal context, in the very same way as the spatial ones. For CSTR point processes, the pair correlation function takes the value of one. Values other than these benchmarks indicate how likely a pair of "extra" points will appear in a CSTR process with the same intensity at the specified locations; i.e., larger values indicate clustering, and smaller values indicate regularity. The pair correlation (Gabriel and Diggle 2009; Møller 2012; González et al. 2016) is defined as

$$g(\xi_1, \xi_2) = \frac{\lambda^{(2)}(\xi_1, \xi_2)}{\lambda(\xi_1)\lambda(\xi_2)}, \qquad \xi_1, \xi_2 \in W \times T. \tag{7}$$

Second-order intensity reweighted stationarity (SOIRS) and isotropy are defined in the same vein as the spatial counterparts. For the estimation of the pair correlation function, consider a point pattern $X = \{(\mathbf{u}_i, v_i)\}_{i=1}^n$ with n points, the spatio-temporal pair correlation function (Eq. (7)) may be estimated by

$$\hat{g}(r,t) = \frac{1}{4\pi r}\sum_{i=1}^{n}\sum_{j\neq i}\frac{\kappa_\epsilon(\|\mathbf{u}_i-\mathbf{u}_j\|-r)\kappa'_\delta(|v_i-v_j|-t)}{\hat{\lambda}(\mathbf{u}_i,v_i)\,\hat{\lambda}\left(\mathbf{u}_j,v_j\right)w_{ij}}, \quad r\in[\epsilon,r_0], t\in[\delta,t_0],$$

where κ_ϵ and κ'_δ are one-dimensional kernel functions with spatial and temporal bandwidths ϵ and δ, respectively; w_{ij} represent edge-correction weights (González et al. 2016). Selecting the bandwidths is often complicated because the estimation depends on those parameters, which means a wrong decision could be disastrous. As the second-order descriptors are distance-based, we can take advantage of some techniques for bandwidth selection. For example, for the temporal kernel, we use the `dpik()` function from the KernSmooth package (González and Moraga 2022b), which uses the method proposed by (Sheather and Jones 1991) to estimate a suitable bandwidth,

```
library(KernSmooth)
dt <- dist(unique(Times))
ht <- dpik(dt, kernel = "epanech", gridsize = 50, scalest = "iqr")
ht <- 2 * ht
```

Note that we set Epanechnikov's kernel as a default because the kernel selection is not as important as the bandwidth. We get a temporal bandwidth of 13.5 days (the output of the `dpik()` function). For the spatial bandwidth and upper bound r_0, we use the same as obtained for the spatial summary functions. For the temporal upper bound t_0, we set the 15% of the maximum temporal distances,

```
t0 <- 0.15 * max(dt)
```

We use the stpp package to estimate the descriptor. We first create data in spatio-temporal point format using the command `as.3dpoints()`; then, we create a two-column matrix to specify the polygonal region containing the points and set the spatial and temporal domains for the descriptor. Finally, we use the `PCFhat()` function to compute the estimate of the pair correlation function,

```
library(stpp)
FMD <- as.3dpoints(PP$x, PP$y, PP$marks$time)
s.region <- as.matrix(data.frame(x = PP$window$bdry[[1]]$x,
                                 y = PP$window$bdry[[1]]$y))
hs <- bwG
u1 <- seq(hs, r0, length.out = 71)[-1]
v1 <- seq(ht, t0, length.out = 71)[-1]
```

```
# Time consuming: This takes roughly 3.87 hours to be executed
g <- PCFhat(xyt = FMD, s.region = s.region, t.region = range(Times),
            lambda = lambda, dist = u1, times = v1, ks = "epanech",
            kt = "epanech", hs = hs, ht = ht)
```

To visualise the descriptor, we use the `persp3D()` function of the plot3D package,

```
library(plot3D)
par(mar = c(0,0,0,0))
persp3D(x = u1, y = v1, z = g$pcf, facets = NA, curtain = F, col =  viridis(200),
  colkey = F, bty = "g", pch = 20, cex = 1.5, theta = 40, phi = 5,
  border = NA, ticktype = "detailed", cex.axis = 0.5,  zlab = "",
  xlab = "spatial distances", ylab = "temporal distances")
```

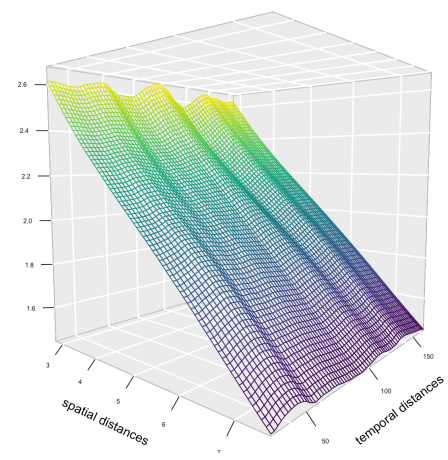

Figure 6: Spatio-temporal pair correlation function estimate of non-specific gastrointestinal disease data.

The $\hat{g}(r,t)$ surface, shown in Figure 6, describes the spatio-temporal interaction structure of the disease; it shows clustering ($\hat{g}(r,t)\geq 1$) for all the distances at all times with the appearance of high clustering at small distances (up to 5km) that moves towards complete spatio-temporal randomness (CSTR) for more considerable distances. We then can suspect from this behaviour that the point pattern is only clustered in the spatial dimension, i.e., the interaction between gastrointestinal reports increases as long as they are close in space, but the clustering in time seems uniform along with the selected periods. We can also conclude that while the first-order intensity is not separable, perhaps the second-order one is. Note that the bandwidth selection explains the waves of the pair correlation function in the temporal dimension; narrow bandwidths produce spurious bumps in the estimate.

One of the most used second-order descriptor is the inhomogeneous spatio-temporal K-function Møller (2012). Under the SOIRS assumption, if $g(r,t)$ is the pair correlation function, then the K-function is defined as

$$K(r,t) = \int_{\mathbb{R}^2}\int_{\mathbb{R}} \mathbf{1}[||\mathbf{u}||\leq r, |s|\leq t]g(||\mathbf{u}||,|s|)\mathrm{d}\mathbf{u}\mathrm{d}s, \quad \text{for } r>0, t>0.$$

In the homogeneous case, $K(r,t)$ is simply the expected number of additional points within distance r and time lag t from the origin, given that the point pattern has a point at the origin. For a CSTR point processes $K(r,t)=\pi r^2 t$. Considering a spatio-temporal point pattern X again, an estimator of $K(r,t)$ is

$$\hat{K}(r,t) = \frac{1}{|W\times T|}\sum_{i=1}^{n}\sum_{j\neq i}\frac{\mathbf{1}\left[\|\mathbf{u}_i-\mathbf{u}_j\|\leq r\right]\mathbf{1}\left[|v_i-v_j|\leq t\right]}{\hat{\lambda}(\mathbf{u}_i,v_i)\,\hat{\lambda}\left(\mathbf{u}_j,v_j\right)w_{ij}}.$$

There is an alternative estimator $\hat{K}^*(r,t)$ matching the original definition of (Gabriel 2014) that into account the past of the process. To compute the K-function we employ the `STIKhat()` function from the stpp package,

```
u <- seq(0, r0, length.out = 71)
v <- seq(0, t0, length.out = 71)
```

```
# Time consuming: This takes roughly 55 minutes to be executed
stik <- STIKhat(xyt = FMD, s.region = s.region, t.region = range(Times),
                lambda = lambda, dist = u, times = v, infectious = F)
```

The parameter `infectious` controls the type of estimator, when it is set `FALSE`, past and future events are considered for the estimation. In this equation, the points $\xi_i=(\mathbf{u}_i,v_i)$ are ordered so that $v_i<v_{i+1}$ (ties are broken by random unrounding whenever necessary). To deal with temporal edge-effects, for each $t\in T=[T_0,T_1]$, n_t is the number of events for which $v_i\leq T_1-t$. For spatial edge effects, the `STIKhat()` function employs Ripley's method (Gabriel and Diggle 2009). For plotting the spatio-temporal K-function we write,

```
KS <- stik$Khat - stik$Ktheo
par(mar = c(0,0,0,0))
persp3D(x = u[-1], y = v[-1], z = KS, facets = NA, zlab = "",
  curtain = F, col =  viridis(100), colkey = F, bty = "g", pch = 20,
  cex = 1.5, theta = 40, phi = 5, border = NA, ticktype = "detailed",
  cex.axis = 0.5, xlab = "spatial distances", ylab = "temporal distances")
```

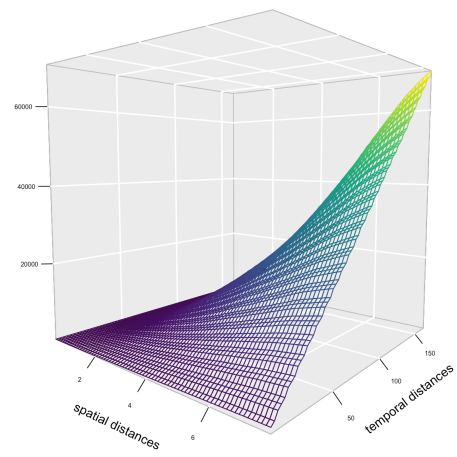

Figure 7: Spatio-temporal pair correlation function estimate of non-specific gastrointestinal disease data.

Figure 7 shows the estimated K-function. Since the K-function is centred on its theoretical value $(2\pi r^2 t)$, we can easily interpret it; positive deviations suggest clustering, while negative deviations represent regularity or inhibition. In our case, we can see positive deviations that become larger as the spatial and temporal distances grow together. Therefore, as time passes and spatial distances become larger, the level of clustering also increases in our point pattern, at least in the range we have considered, that is, up to 7km in space and up to 5 months in time.

4 Discussion

Throughout this paper, we have shown how to analyse complex spatial and spatio-temporal point patterns through various techniques that aim to capture their first and second-order characteristics using several R packages. Specifically, we have toured very recent techniques such as variable bandwidths, relative risk inference, and composite hypotheses. The spatial methods presented are well-investigated and documented and can be easily applied using R. We have seen that some procedures, such as estimating p-values in composite hypotheses, require severe computational resources. The approximate computing time, including all the routines and set parameters suggested throughout the paper, is approximately six hours in a 3 GHz 6-Core Intel Core i5 iMac computer. This time depends on several factors, such as the number of simulations and the precision of desired outputs. It is mainly due to the number of points in the point pattern; the more data points, the more computational complexity, especially in the case of second-order descriptors, where pairwise calculations are required. Some R packages such as foreach can help accelerate the computations, even though these methods may quickly become unfeasible when the datasets have many points.

We have presented the most straightforward methods and the most widely known in the literature; we have done this, above all, because we intended to provide a solid and approachable learning resource. However, there are also alternative methods to deal with some of the descriptors we analyse in this document; for example, González and Moraga (2022a) presents a methodology to accurately and adaptively estimate the spatio-temporal intensity functions; this methodology can be applied using the kernstadapt package.

The analyses presented here can benefit future researchers who have initial questions about spatio-temporal point process data; these results can serve, for instance, as input for possible statistical models. We have not delved into statistical modelling and inference since many models exist in the spatio-temporal point processes literature. These statistical models can incorporate current knowledge of the phenomenon and many scientific considerations. Different models can be proposed and compared to decide which variables significantly influence the summary descriptors, making the results more informative (Baddeley et al. 2015). For example, one of the most flexible families of models with the most methodological advantages is the family of log-Gaussian point processes. Log-Gaussian Cox processes are models for point patterns that intend to capture the environmental variability (Diggle et al. 2013). They are spatio-temporal Poisson processes conditional on the realisation $\lambda(\mathbf{u}, v)$ of a stochastic intensity function $\Lambda(\mathbf{u}, v)$, $(\mathbf{u}, v) \in \mathbb{R}^2 \times \mathbb{R}$. In the SOIRS case, a convenient representation of the intensity is

$$\log\{\Lambda(\mathbf{u}, v)\} = \lambda(\mathbf{u}, v) S(\mathbf{u}, v),$$

where $S(\mathbf{u}, v)$ is a Gaussian stationary process with expectation 1 and covariance function $\gamma(r, t) = \sigma^2 s(r, t)$, where σ^2 is the variance of $S(\mathbf{u}, v)$ and $s(\cdot, \cdot)$ is a spatio-temporal correlation function, where r and t are spatial and temporal distances. Inference uses a discretised version of $S(\mathbf{u}, v)$, namely $\mathcal{S}$, defined on a regular grid, where observations X are the cell counts, and the intensity of the process is assumed constant within each cell (Taylor and Diggle 2014). $\mathcal{S}$ is a finite collection of Gaussian random variables, so its properties imply a multivariate Gaussian density. The covariance matrix elements are calculated by evaluating $\gamma(r, t)$ at the centroids of the cells. Predictive inference about $\mathcal{S}$ requires samples from the conditional distribution of the latent field given the observations, namely $[\mathcal{S}|X]$. For estimating the parameters of a space-time log-Gaussian Cox process, one can choose between computationally feasible methods, such as those based on moments or more demanding methods based on likelihood (Diggle et al. 2013). Inference can be achieved for example, by using Markov chain Monte Carlo (MCMC) for which the lgcp package can be used, and by using integrated nested Laplace approximation (INLA) with the inla and inlabru packages.

To conclude, the methods presented here are primarily concerned with understanding the probability distribution that generate points observed in space or time. The study of such a distribution involves many routines, concepts and contexts that can help answer scientific questions and make decisions based on the data or inferences about them. The inclusion of technological advancements in R has led to improved efficiency and speed in calculations. Additionally, R offers a versatile and reliable environment for implementing statistical procedures and creating graphical visualisations. Also, modern developments in spatial statistics related to point patterns have been made possible thanks to the widespread availability of reproducible R scripts for analyses. In this paper, we have made extensive use of the ability to apply, extend and modify procedures in R to analyse spatial and spatio-temporal point patterns in the context of epidemiology. These approaches are also helpful for analysing point pattern data that arise in various fields such as climatology, ecology or seismology.

Supplementary materials

Supplementary materials are available in addition to this article. It can be downloaded at RJ-2023-025.zip

CRAN packages used

spatial, splancs, sp, sf, spatstat, spatstat.utils, spatstat.sparse, spatstat.data, spatstat.geom, spatstat.random, spatstat.explore, spatstat.model, spatstat.linnet, smacpod, sparr, spatgraphs, stpp, PtProcess, lgcp, inlabru, GET, foreach, doParallel, viridis, ggplot2, kernstadapt, KernSmooth, plot3D

CRAN Task Views implied by cited packages

HighPerformanceComputing, MixedModels, Phylogenetics, Spatial, SpatioTemporal, Survival, TeachingStatistics

References

F. E. Bachl, F. Lindgren, D. L. Borchers and J. B. Illian. inlabru: An R package for Bayesian spatial modelling from ecological survey data. *Methods in Ecology and Evolution*, 10(6): 760–766, 2019.

A. Baddeley, A. Hardegen, T. Lawrence, R. K. Milne, G. Nair and S. Rakshit. On two-stage Monte Carlo tests of composite hypotheses. *Computational Statistics & Data Analysis*, 114: 75–87, 2017.

A. Baddeley, E. Rubak and R. Turner. *Spatial point patterns: Methodology and applications with r*. CRC Press, Boca Raton, Florida, 2015.

A. Baddeley and R. Turner. spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6): 1–42, 2005.

J. Besag. Contribution to the discussion of dr Ripley's paper. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39: 193–195, 1977.

R. S. Bivand, E. Pebesma and V. Gómez-Rubio. *Applied spatial data analysis with R*. Second Springer, New York, 2013.

E. Choi and P. Hall. Nonparametric approach to analysis of space-time data on earthquake occurrences. *Journal of Computational and Graphical Statistics*, 8: 733–748, 1999.

D. Daley and D. Vere-Jones. *An introduction to the theory of point processes: Volume i: Elementary theory and methods*. Second Springer-Verlag, New York, 2003.

T. M. Davies and A. Baddeley. Fast computation of spatially adaptive kernel estimates. *Statistics and Computing*, 28(4): 937–956, 2018.

T. M. Davies, J. C. Marshall and M. L. Hazelton. Tutorial on kernel estimation of continuous spatial and spatiotemporal relative risk. *Statistics in Medicine*, 37(7): 1191–1221, 2018.

P. J. Diggle. A kernel method for smoothing point process data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 34(2): 138–147, 1985.

P. J. Diggle. *Statistical analysis of spatial and spatio-temporal point patterns*. Third CRC Press, Boca Raton, Florida, 2013.

P. J. Diggle, P. Moraga, B. Rowlingson and B. Taylor. Spatial and spatio-temporal log-Gaussian Cox processes: Extending the geostatistical paradigm. *Statistical Science*, 28: 542–563, 2013.

P. Diggle, L. Knorr-Held, R. B, T. Su, P. Hawtin and T. Bryant. On-line monitoring of public health surveillance data. In *Monitoring the health of populations: Statistical principlesand methods for public health surveillance*, Eds B. R. and S. D. F. pages. 233–266 2003. Oxford University Press.

E. Gabriel. Estimating second-order characteristics of inhomogeneous spatio-temporal point processes. *Methodology and Computing in Applied Probability*, 16: 411–431, 2014.

E. Gabriel and P. J. Diggle. Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica*, 63(1): 43–51, 2009.

E. Gabriel, B. Rowlingson and P. J. Diggle. stpp: An `R` package for plotting, simulating and analyzing spatio-temporal point patterns. *Journal of Statistical Software*, 53(2)(2): 1–29, 2013.

S. Garnier, Simon, N. Ross, B. Rudis, M. Sciaini, A. Pedro and C. Scherer. *`viridis(Lite)` - colorblind-friendly color maps for `R`*. 2023. URL https://sjmgarnier.github.io/viridis/. viridis package version 0.6.3.

M. Ghorbani, N. Vafaei, J. Dvořák and M. Myllymäki. Testing the first-order separability hypothesis for spatio-temporal point patterns. *Computational Statistics & Data Analysis*, 161: 107245, 2021.

J. A. González, U. Hahn and J. Mateu. Analysis of tornado reports through replicated spatiotemporal point patterns. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 69(1): 3–23, 2019.

J. A. González and P. Moraga. An adaptive kernel estimator for the intensity function of spatio-temporal point processes. 2022a. DOI 10.48550/ARXIV.2208.12026.

J. A. González and P. Moraga. `kernstadapt`: Spatio-temporal adaptive kernel estimators for intensities. 2022b. URL https://CRAN.R-project.org/package=kernstadapt. R package version 0.0.2.

J. A. González, F. J. Rodríguez-Cortés, O. Cronie and J. Mateu. Spatio-temporal point process statistics: A review. *Spatial Statistics*, 18: 505–544, 2016.

D. Harte. PtProcess: An R package for modelling marked point processes indexed by time. *Journal of Statistical Software*, 35(8): 1–32, 2010.

M. L. Hazelton and T. M. Davies. Inference based on kernel estimates of the relative risk function in geographical epidemiology. *Biometrical Journal*, 51(1): 98–109, 2009.

J. Illian, P. A. Penttinen, H. Stoyan and D. Stoyan. *Statistical analysis and modelling of spatial point patterns*. Wiley, 2008.

A. Jalilian and R. Waagepetersen. Fast bandwidth selection for estimation of the pair correlation function. *Journal of Statistical Computation and Simulation*, 88(10): 2001–2011, 2018.

J. E. Kelsall and P. J. Diggle. Kernel estimation of relative risk. *Bernoulli*, 1(1/2): 3–16, 1995.

Microsoft and S. Weston. *`doParallel`: Foreach parallel adaptor for the 'parallel' package*. 2022a. URL https://CRAN.R-project.org/package=doParallel. R package version 1.0.17.

Microsoft and S. Weston. *`foreach`: Provides foreach looping construct*. 2022b. URL https://CRAN.R-project.org/package=foreach. R package version 1.5.2.

M. Møller J. And Ghorbani. Aspects of second-order analysis of structured inhomogeneous spatio-temporal point processes. *Statistica Neerlandica*, 66: 472–491, 2012.

P. Moraga. *Geospatial health data: Modeling and visualization with R-INLA and shiny*. Chapman & Hall/CRC, Boca Raton, Florida, 2019.

P. Moraga. Species distribution modeling using spatial point processes: A case study of sloth occurrence in Costa Rica. *The R Journal*, 12(2): 293–310, 2021.

P. Moraga and F. Montes. Detection of spatial disease clusters with LISA functions. *Statistics in Medicine*, 30(10): 1057–1071, 2011.

M. Myllymäki and T. Mrkvička. GET: Global envelopes in R. *arXiv:1911.06583 [stat.ME]*, 2020. URL https://arxiv.org/abs/1911.06583.

M. Myllymäki, T. Mrkvička, P. Grabarnik, H. Seijo and U. Hahn. Global envelope tests for spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2): 381–404, 2017.

E. Pebesma. Simple features for R: Standardized support for spatial vector data. *The R Journal*, 10(1): 439–446, 2018.

E. J. Pebesma and R. S. B. Classes and methods for spatial data in R. *R News*, 5(2): 9–13, 2005.

R Core Team. *`R`: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2023. URL https://www.R-project.org.

B. D. Ripley. Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(2): 172–212, 1977.

A. Rodrigues and P. J. Diggle. Bayesian estimation and prediction for inhomogeneous spatiotemporal log-Gaussian Cox processes using low-rank models, with application to criminal surveillance. *Journal of the American Statistical Association*, 107(497): 93–101, 2012.

B. S. Rowlingson and P. J. Diggle. splancs: Spatial point pattern analysis code in S-PLUS. *Computers & Geosciences*, 19(5): 627–655, 1993.

H. Rue, S. Martino and N. Chopin. Approximate bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2): 319–392, 2009.

D. W. Scott. *Multivariate density estimation: Theory, practice, and visualization*. Second John Wiley & Sons, 2015.

S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 53(3): 683–690, 1991.

B. W. Silverman. *Density estimation for statistics and data analysis*. CRC press, Boca Raton, Florida, 1986.

B. M. Taylor, T. M. Davies, B. S. Rowlingson and P. J. Diggle. Lgcp: An R package for inference with spatial and spatio-temporal log-Gaussian Cox processes. *Journal of Statistical Software*, 52: 1–40, 2013.

B. Taylor and P. Diggle. INLA or MCMC? A tutorial and comparative evaluation for spatial prediction in log-Gaussian Cox processes. *Journal of Statistical Computation and Simulation*, 84(10): 2266–2284, 2014.

R. Turner. Point patterns of forest fire locations. *Environmental and Ecological Statistics*, 16(2): 197–223, 2009.

W. N. Venables and B. D. Ripley. *Modern applied statistics with s*. Fourth Springer-Verlag, New York, 2002.

H. Wässle, B. B. Boycott and R.-B. Illing. Morphology and mosaic of on-and off-beta cells in the cat retina and some functional considerations. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 212(1187): 177–195, 1981.

H. Wickham. *`ggplot2`: Elegant graphics for data analysis*. Second Springer-Verlag, New York, 2016.

K. Y. Wong and D. Stoyan. Poles of pair correlation functions: When they are real? *Annals of the Institute of Statistical Mathematics*, 73: 425–440, 2021.

Reuse

Text and figures are licensed under Creative Commons Attribution CC BY 4.0. The figures that have been reused from other sources don't fall under this license and can be recognized by a note in their caption: "Figure from ...".

Citation

For attribution, please cite this work as

```
González & Moraga, "Non-Parametric Analysis of Spatial and Spatio-Temporal Point Patterns", The R Journal, 2023
```

BibTeX citation

```
@article{RJ-2023-025,
  author = {González, Jonatan A. and Moraga, Paula},
  title = {Non-Parametric Analysis of Spatial and Spatio
  journal = {The R Journal},
  year = {2023},
  note = {https://doi.org/10.32614/RJ-2023-025},
  doi = {10.32614/RJ-2023-025},
  volume = {15},
  issue = {1},
  issn = {2073-4859},
  pages = {65-82}
}
```

© The R Foundation, web

```
-Temporal Point Patterns},
```

page contact.