B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993. [p]

M. Febrero-Bande and M. Oviedo de la Fuente. Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, 51(4):1–28, 2012. [p]

F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer-Verlag, New York, 2006. [p]

L. Horvath and P. Kokoszka. *Inference for Functional Data with Applications*. Springer-Verlag, New York, 2012. [p]

F. Ieva and A. M. Paganoni. Depth measures for multivariate functional data. *Communication in Statistics - Theory and Methods*, 42(7):1265 – 1276, 2013. [p]

F. Ieva and A. M. Paganoni. Component-wise outlier detection methods for robustifying multivariate functional samples. *Statistical Papers*, 2017. URL https://doi.org/10.1007/s00362-017-0953-1. [p]

F. Ieva, A. M. Paganoni, D. Pigoli, and V. Vitelli. Multivariate functional clustering for the morphological analysis of ecg curves. *Journal of the Royal Statistical Society C*, 62(3):401 – 418, 2013. [p]

F. Ieva, F. Palma, and J. Romo. Bootstrap-based inference for dependence in multivariate functional data. MOX Report 30/2018, Politecnico di Milano, 2018. URL https://www.mate.polimi.it/biblioteca/add/qmox/30-2018.pdf. [p]

P. Kokoszka and M. Reimherr. *Introduction to Functional Data Analysis*. Chapman & Hall, 2017. [p]

R. Liu and K. Singh. A quality index based on data depth and multivariate rank tests. *Journal of the American Statistical Association*, 88(421):252 – 260, 1993. [p]

R. Liu, J. M. Parelius, and K. Singh. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):783–858, 1999. [p]

S. Lopez-Pintado, Y. Sun, and M. G. Genton. Simplicial band depth for multivariate functional data. *Advances in Data Analysis and Classification*, 8:321–338, 2014. [p]

S. López-Pintado and J. Romo. Depth-based inference for functional data. *Computational Statistics & Data Analysis*, 51(10):4957–4968, 2007. [p]

S. López-Pintado and J. Romo. On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718 – 734, 2009. [p]

S. López-Pintado and J. Romo. A half-region depth for functional data. *Computational Statistics & Data Analysis*, 55(4):1679–1695, 2011. [p]

R. A. Maronna and R. H. Zamar. Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics*, 44(4):307–317, 2002. [p]

J. O. Ramsay and B. W. Silverman. *Applied Functional Data Analysis: Methods and Case Studies*. Springer-Verlag, New York, 2002. [p]

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer-Verlag, New York, 2nd edition, 2005. [p]

J. O. Ramsay, G. Hooker, and S. Graves. *Functional Data Analysis with R and MATLAB*. Springer-Verlag, 1st edition, 2009. [p]

J. O. Ramsay, H. Wickham, S. Graves, and G. Hooker. *Fda: Functional Data Analysis*, 2014. URL https://CRAN.R-project.org/package=fda. R package version 2.4.4. [p]

C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. [p]

H. L. Shang and R. Hyndman. *Rainbow: Bagplots, Boxplots and Rainbow Plots for Functional Data*, 2019. URL https://CRAN.R-project.org/package=rainbow. R package version 3.6. [p]

Y. Sun and M. G. Genton. Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2): 316–334, 2011. [p]

Y. Sun and M. G. Genton. Adjusted functional boxplots for spatio-temporal data visualization and outlier detection. *Environmetrics*, 23(1):54–64, 2012. [p]

N. Tarabelloni, F. Ieva, R. Biasi, and A. M. Paganoni. Use of depth measure for multivariate functional data in disease prediction: An application to electrocardiograph signals. *The international journal of biostatistics*, 11(2):189–201, 2015. [p]

N. Tarabelloni, A. Arribas-Gil, F. Ieva, A. M. Paganoni, and J. Romo. *Roahd: Robust Analysis of High Dimensional Data*, 2017. URL https://CRAN.R-project.org/package=roahd. R package version 1.3. [p]

J. D. Tucker. *Fdasrvf: Elastic Functional Data Analysis*, 2017. URL https://CRAN.R-project.org/package=fdasrvf. R package version 1.8.3. [p]

J. Tuckey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver*, volume 2, pages 523 – 531, 1975. [p]

D. Valencia, J. Romo, and R. Lillo. A kendall correlation coefficient for functional dependence. Technical report, Universidad Carlos III de Madrid, http://EconPapers.repec.org/RePEc:cte:wsrepe:ws133228, 2015a. [p]

D. Valencia, J. Romo, and R. Lillo. Spearman coefficient for functions. Technical report, Universidad Carlos III de Madrid, http://EconPapers.repec.org/RePEc:cte:wsrepe:ws133329, 2015b. [p]

A. Zeileis. CRAN task views. *R News*, 5(1):39–40, 2005. URL https://CRAN.R-project.org/doc/Rnews/. [p]

Y. Zuo and R. Serfling. General notions of statistical depth function. *The Annals of Statistics*, 28(2): 461–482, 2000. [p]

Francesca Ieva
MOX - Department of Mathematics
Politecnico di Milano
Italy
francesca.ieva@polimi.it

Anna Maria Paganoni
MOX - Department of Mathematics
Politecnico di Milano
Italy
anna.paganoni@polimi.it

Juan Romo
Department of Statistics
Universidad Carlos III de Madrid
Spain
juan.romo@uc3m.es

Nicholas Tarabelloni
MOX - Department of Mathematics
Politecnico di Milano
Italy
nicholas.tarabelloni@polimi.it

The IDSpatialStats R Package: Quantifying Spatial Dependence of Infectious Disease Spread

by John R. Giles, Henrik Salje, and Justin Lessler

Abstract Spatial statistics for infectious diseases are important because the spatial and temporal scale over which transmission operates determine the dynamics of disease spread. Many methods for quantifying the distribution and clustering of spatial point patterns have been developed (e.g. *K*-function and pair correlation function) and are routinely applied to infectious disease case occurrence data. However, these methods do not explicitly account for overlapping chains of transmission and require knowledge of the underlying population distribution, which can be limiting when analyzing epidemic case occurrence data. Therefore, we developed two novel spatial statistics that account for these effects to estimate: 1) the mean of the spatial transmission kernel, and 2) the τ-statistic, a measure of global clustering based on pathogen subtype. We briefly introduce these statistics and show how to implement them using the IDSpatialStats R package.

Introduction

The transmission process which drives an epidemic can be characterized by the spatial distance separating linked cases. When these transmission events accumulate over time, they are observed as areas of elevated disease prevalence. Knowledge of the extent of the affected area and where new cases may arise is crucial for many disease control strategies (e.g. ring vaccination, vector control etc). In epidemiology, case occurrence data—(x,y) coordinates with temporal information (t) and other covariates—are often used to understand these types of infectious disease dynamics. These data are typically treated as a generic point process so that they can be described in terms of spatial intensity (expected number of cases per unit area) or clustering due to spatial dependence (covariance in x,y space).

In the broader field of spatial statistics, there are many methods that measure the spatial intensity or clustering of a generic point process on a Cartesian (x,y) coordinate system (Table 1). These methods primarily fall into three categories: first-order first-moment (FOFM), first-order second-moment (FOSM), and second-order second-moment (SOSM). The FOFM measures use quadrature (aggregate counts of points within cells) to quantify intensity of the point pattern continuously over (x,y) space. Packages such as **lgcp** (Taylor et al., 2013, 2015) and **ppmlasso** (Renner and Warton, 2013) allow users to model first-order intensity as a count process using a regressive function of (x,y) coordinates and other covariates. The FOSM measures—Moran's I and Geary's C (Moran, 1950; Geary, 1954)—also use quadrature, but they describe general covariance among cell counts across the (x,y) dimensions. These spatial statistics can be calculated using the **spdep** R package (Bivand and Piras, 2015). The SOSM measures, such as the *K*-function and its non-cumulative analogue, the pair correlation function, quantify clustering among neighboring points. Both the FOSM and SOSM measures are considered global spatial statistics because they describe spatial dependence for the entire study area. However, the SOSM can further describe how spatial dependence changes as a function of distance by comparing the observed intensity of neighboring points within distance d to that expected under complete spatial randomness. The *K*-function and pair correlation function can be calculated using the **ads** (Pélissier and Goreaud, 2015), **spatstat** (Baddeley et al., 2016), and **splancs** Rowlingson and Diggle (2017) R packages.

These classic spatial measures are limited in their ability to describe infectious disease dynamics primarily because they treat case occurrence data as a generic point process. The FOFM and FOSM measures use quadrature, which make them vulnerable to error associated with data aggregation (Robinson, 2009) and the modifiable areal unit problem (Openshaw and Taylor, 1979). The SOSM measures, like the *K*-function and pair correlation function, are more common in epidemiology. However, their statistical interpretation is less intuitive in terms of classic epidemiological quantities of relative disease risk, such as the incidence rate ratio or hazard ratio. Additionally, even the temporal forms of these functions (e.g. the space-time *K*-function) do not capture the typical distances traveled in a single transmission generation as they quantify the overall spatial dependence between all cases, not just those epidemiologically linked. The mean distance between sequential cases in a transmission chain is an important epidemiological quantity because it provides insight into potential mechanisms driving spread as well as helping inform interventions. Therefore, we developed novel measures that build upon concepts in spatial statistics to characterize infectious disease spread using case occurrence

Table 1: A selective list of R packages for the analysis of spatial point pattern data. This list is not exhaustive. Visit the *Spatial* CRAN Task View for a more comprehensive list of resources.

Package	Description	Citation
ads	K function for enclosed point patterns	Pélissier and Goreaud (2015)
DCluster	disease clustering for count data	Gómez-Rubio et al. (2005)
lgcp	modeling point patterns with log-Gaussian Cox processes	Taylor et al. (2013, 2015)
ppmlasso	modeling point patterns with LASSO regularization	Renner and Warton (2013)
SGCS	third order clustering of point processes	Rajala (2017)
sparr	spatial relative risk functions with kernel smoothing	Davies et al. (2011)
spatstat	comprehensive tools for analyzing point patterns in many dimensions	Baddeley et al. (2016)
spdep	classic statistics to test for spatial dependence	Bivand and Piras (2015)
splancs	kernel smoothing and Poisson cluster processes	Rowlingson and Diggle (2017)

data. Importantly, these measures are robust to heterogeneities in the underlying population, and substantial case under-reporting, which is common in epidemiology.

We describe these two measures of spatial dependence for infectious diseases and show how they can be calculated with the **IDSpatialStats** R package in the following three sections. First, we introduce a function which simulates infectious disease spread as a spatial branching process. This function is primarily intended to simulate example datasets for the `est.transdist` family of functions and τ-statistics that use temporal information to indicate linked cases. Second, we demonstrate how to estimate the mean and standard deviation of the spatial transmission kernel (Salje et al., 2016b). Estimating the spatial transmission kernel requires an understanding of the number of transmission generations separating cases at different time points of the epidemic. This method provides a measure of fine-scale spatial dependence between two cases, which can be interpreted as the mean distance between sequential cases in a transmission chain. Third, we describe a measure of global clustering—the τ-statistic—that calculates the relative risk of infection given some criteria to identify cases closely related along a chain of transmission (Lessler et al., 2016). The τ-statistic is a global clustering statistic—like the K-function and pair correlation function—that provides an overall measure of clustering for the entire course of an outbreak. Depending on the parameterization, the τ-statistic represents the odds of observing another case with distance d of an infected case compared with either the underlying population or other pathogen types. The following sections contain a brief introduction to each statistic to provide context to the code implementation—for more detailed description of each statistic, see Lessler et al. (2016) and Salje et al. (2016b).

We have implemented these tools in the **IDSpatialStats** R package version 0.3.5 and above. The latest stable release depends on the `doParallel` and `foreach` packages (Microsoft and Weston, 2017; Corporation and Weston, 2018) and can be downloaded from CRAN. A development version of the package is also available on Github at `https://github.com/HopkinsIDD/IDSpatialStats`.

Simulating spatial disease spread

We use a stochastic spatial branching process to simulate epidemiological data in the `sim.epidemic` function. Simulations begin with an index case at $(x, y, t) = (0, 0, 0)$ and transmission events that link two cases follow according to a random Markov process in (x, y) space (i.e. Brownian motion). The number of transmission events occurs according to a Poisson distribution, with its mean and variance set to the effective reproduction number R of the infecting pathogen. The spatial distance traversed by each transmission event is given by a user specified probability distribution which serves as the dispersal kernel function. When specifying the dispersal kernel, the `trans.kern.func` argument expects a list object containing a probability distribution function and its named arguments. For example, to simulate an epidemic where transmission typically occurs at the local level, but long distance transmission events sometimes occur, an exponential transmission kernel might be used because of its long tail. Alternatively, if transmission is expected to consistently occur within a given range, then a normal kernel may be more appropriate.

In simulations where the basic reproductive number R_0 is used to define a constant R-value and $R_0 > 1$, the number of cases will continue to increase with each time step. This effect may not be appropriate when simulating settings where intervention efforts or depletion of susceptible individuals causes heterogeneity in R over the course of the epidemic. Thus, the `sim.epidemic`

function accepts either a scalar value for a constant R value or a vector of R values with a length equal to `tot.generations`, allowing simulations with a variable R value, as shown in the following R code.

```
# Epidemic simulations with variable R value
R1 <- 2
R2 <- 0.5
tot.gen <- 12
R.vec <- seq(R1, R2, (R2 - R1)/(tot.gen - 1))
dist.func <- alist(n=1, a=1/100, rexp(n, a))
sim <- sim.epidemic(R=R.vec, gen.t.mean=7, gen.t.sd=2, min.cases=100,
                    tot.generations=tot.gen, trans.kern.func=dist.func)

head(sim, n=4)
          x           y t
1   0.00000   0.000000 0
2  24.46125   3.280527 3
3 -60.73475 184.885784 7
4 -12.79933 -57.798696 4

sim.plot(sim)
```

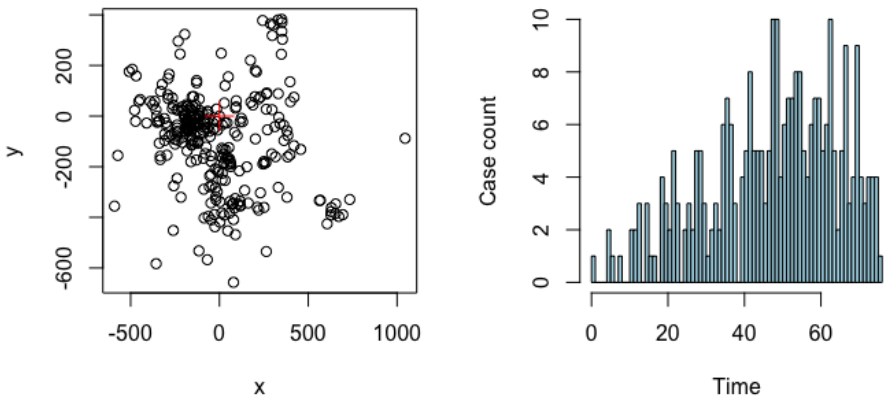

Figure 1: Left: the spatial distribution of simulated cases with the red cross showing the index case. Right: the epidemic curve for a simulation with an R value decreasing from 2 to 0.5 over the course of the epidemic.

The mean transmission distance

In Salje et al. (2016b), we introduced a method to estimate the mean and standard deviation of the spatial transmission kernel using case occurrence data. These data include location (x, y) and onset time t of each case (case times) and the infecting pathogen's generation time $g(x)$. To estimate these spatial statistics, we use the Wallinga-Teunis (WT) method (Wallinga and Teunis, 2004) to probabilistically estimate the number of transmission events required to link two cases, denoted as θ. In settings where a phylogenetic model or contact tracing provide information on transmission pathways, the spatial kernel can be empirically estimated using the distribution of observed distances among all linked cases. The mean and standard deviation of this kernel can then be calculated for any time interval between t_1 and t_2 to give $\mu_t^{obs}(t_1, t_2)$, with the assumption that the number of transmission events separating all case pairs is homogeneous ($\theta = 1$). When data that indicate case linkage are lacking, this assumption is incorrect because the distance between two cases depends on the number of transmission events separating them. In this case, the mean transmission distance at each time interval μ_t must be estimated as a weighted mean:

$$\mu_t(t_1, t_2, \mu_k, \sigma_k) = \sum_i w(\theta = i, t_1, t_2) \cdot \mu_a(\theta = i, \mu_k, \sigma_k).$$

Where, $w(\theta = i, t_1, t_2)$ gives the weight for each of the i elements of θ and the second term $\mu_a(\theta = i, \mu_k, \sigma_k)$ gives the mean distance separating case pairs that are linked by the ith value of θ.

We have implemented four nested functions that are used to estimate $w(\theta = i, t_1, t_2)$ and describe them briefly below. Listed in order, they are comprised of `est.wt.matrix.weights`, `est.wt.matrix`, `get.transdist.theta`, and `est.transdist.theta.weights`. Although, these functions are documented separately, they are all driven by the `est.transdist` family of functions and do not need to be run manually unless desired.

Wallinga-Teunis matrices

The `est.wt.matrix.weights` function builds upon code from the `R0` package (Obadia et al., 2012) to calculate the basic WT matrix (Wallinga and Teunis, 2004). This matrix gives the probability that a case at time t_i (rows) infected a case at time t_j (columns), i.e. $\theta = 1$, based on the generation time distribution of the pathogen $g(x)$. For an epidemic with t unique case times, `est.wt.matrix.weights` gives a $T \times T$ matrix.

The `est.wt.matrix` function produces a WT type matrix for all infector-infectee case pairs. Given the WT matrix produced by `est.wt.matrix.weights` and total case count n, this function calculates an $n \times n$ matrix giving the probability that case i (rows) infected case j (columns). The WT matrix object can be handed directly to `est.wt.matrix` via the `basic.wt.weights` argument, or if this argument is `NULL`, the `est.wt.matrix.weights` function is called automatically.

```
# Calculating Wallinga-Teunis matrices
case.times <- c(1,2,2,3,3) # times each case occurs
g.x <- c(0, 2/3, 1/3, 0, 0) # hypothetical generation time of a pathogen

mat.wts <- est.wt.matrix.weights(case.times=case.times, gen.t.dist=g.x)

# Calculate infector-infectee Wallinga-Teunis matrix
wt.mat1 <- est.wt.matrix(case.times=case.times, gen.t.dist=g.x,
                         basic.wt.weights=mat.wts)
wt.mat2 <- est.wt.matrix(case.times=case.times, gen.t.dist=g.x)

identical(wt.mat1, wt.mat2) # the two methods are equivalent
[1] TRUE
```

Estimation of θ weights

The `get.transdist.theta` function estimates the number of transmission events θ separating pairs of cases using the probabilities in the infector-infectee WT matrix produced by the `est.wt.matrix` function. Sampling all possible transmission trees is impractical for most datasets, so this function constructs a transmission tree by randomly selecting the infector of each case in the epidemic and then θ is determined by finding the product of all probabilities in the chain of transmission that link the randomly sampled case pairs.

The object `theta.wts` (in the code segment below) contains a three-dimensional array [i,j,k], where the rows i and columns j represent unique case times and the third dimension k is the number of transmission events θ. Each cell gives the probability that two cases occurring at times i and j are connected by θ transmission events in the randomly sampled transmission tree. Probabilities in each [i,j,$\cdot$] row are normalized across all θ values. The `get.transdist.theta` function samples a single randomized transmission tree from the epidemic data, therefore we want to simulate many iterations of this random sampling to get a better estimate of the true distribution of θ.

The `est.transdist.theta.weights` function estimates the distribution of θ across all t_i and t_j combinations by simulating many iterations of transmission trees using the `get.transdist.theta` function. Its output is the same as the `get.transdist.theta` function, however, it represents the normalized probabilities after `n.rep` number of simulations.

```
# Estimate theta weights
case.times <- c(1,2,2,3,3) # times each case occurs
g.x <- c(0, 2/3, 1/3, 0, 0) # hypothetical generation time distribution of a pathogen
gen.time <- 1 # mean generation time
n.gen <- round((max(case.times) - min(case.times)) / gen.time) + 1 # total generations

# Calculate infector-infectee Wallinga-Teunis matrix
wt.mat <- est.wt.matrix(case.times=case.times, gen.t.dist=g.x)
```

```
# Estimated theta weights from five randomized transmission trees
theta.wts <- est.transdist.theta.weights(case.times=case.times, n.rep=5,
                                         gen.t.mean=gen.time, t1=0, t.density=g.x)
theta.wts[,,1]
      [,1]   [,2] [,3]
[1,] 0.000    NaN  NaN
[2,] 0.625 0.0000  NaN
[3,] 0.000 0.4375    0
```

Estimating mean of transmission kernel

To estimate the mean transmission distance over the duration of the epidemic we must use the observed distances between case pairs given the time they occurred $\mu_t^{obs}(t_i, t_j)$ and combine them into an overall estimate of the mean of the transmission kernel μ_k. The workhorse function `est.trandsdist` estimates the overall mean μ_k and standard deviation σ_k of the kernel. This function first calls the `est.wt.matrix.weights`, `est.wt.matrix`, `get.transdist.theta`, and `est.transdist.theta.weights` functions described above to estimate the distribution of θ across all case pairs and then calculates each of the weights $w(\theta = i, t_1, t_2)$. The weights are calculated as the proportion of all case pairs occurring at t_i and t_j that are separated by each estimated θ over all simulations:

$$\hat{w}(\theta = i, t_1, t_2) = \frac{\sum_{k=1}^{N_{sim}} \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{I}_1(t_i = t_1, t_j = t_2, \Theta_{ij} = \theta)}{N_{sim} \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{I}_2(t_i = t_1, t_j = t_2)}.$$

Here, the functions $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ indicate if two cases occurred at time t_i and t_j and were linked by θ transmission events, or if they just occurred at t_i and t_j respectively. In words this can be written as:

$$\hat{w}(\theta = i, t_1, t_2) = \frac{\text{Total cases at } t_1 \text{ and } t_2 \text{ across all simulations separated by } \theta \text{ transmission events}}{\text{Total cases at } t_1 \text{ and } t_2 \text{ across all simulations}}.$$

Once the weights of the θ values are estimated, the `est.transdist` function calculates μ_k and σ_k as the average weighted estimate over all combinations of t_i and t_j. If we now let k index the vector of θ values, then:

$$\hat{\mu}_k = \hat{\sigma}_k = \frac{1}{\sum_i \sum_j n_{ij}} \sum_i \sum_j \frac{2 \cdot \mu_t^{obs}(t_i, t_j) \cdot n_{ij}}{\sum_k \hat{w}(\theta = k, t_i, t_j) \cdot \sqrt{2\pi k}}.$$

For a derivation of these equations, see sections 2.3 and 2.4 of Salje et al. (2016b).

The `est.trandist` function requires case occurrence data—a matrix with three columns [*x*,*y*,*t*]—and the mean and standard deviation of the infecting pathogen's generation time (for calculating WT matrices) as input. The function returns estimates of μ_k and σ_k of the spatial transmission kernel. These estimates are made under the assumption that $\mu_k = \sigma_k$, so the upper bound of $\hat{\mu}_k$ and $\hat{\sigma}_k$ are also provided for when this assumption is violated. Bound estimates are equal to $\sqrt{2}$ times the values estimated under the $\mu_k = \sigma_k$ assumption (see section 2.5 of Salje et al. (2016b)). Additional constraints on the estimation of μ_k and σ_k can be defined in the remaining arguments, such as the time step in which the analysis should begin (`t1`), the maximum number of time steps (`max.sep`) and maximum spatial distance (`max.dist`) to consider when estimating θ, and the number of randomized transmission tree simulations to run (`n.transtree.reps`).

To estimate the uncertainty around $\hat{\mu}_k$ due to sampling or observation error, we have implemented a wrapper function called `est.transdist.bootstrap.ci` that performs bootstrap iterations using the `est.transdist` function. Upon each iteration, the epidemiological data are resampled with replacement and μ_k is re-estimated. The `est.transdist.bootstrap.ci` function contains all the same arguments as the `est.transdist` function, with additional arguments defining the number of bootstrapped iterations to perform, the high and low boundaries of the desired confidence interval, and options for running the bootstrap analysis in parallel.

When parallel computation is enabled (the default is `parallel = FALSE`), the function uses the `makeCluster()` function of the `parallel` package to make the appropriate cluster type for the operating system of the local machine (SOCK cluster for Windows or a Fork cluster for Unix-like machines). The cluster is then registered as the parallel backend for the `foreach` package, which is used to run the bootstraps in parallel. The user can define the number of cores to use when running in parallel using the `n.cores` argument. If `parallel = TRUE` and `n.cores = NULL`, the function will use half the total cores on the local machine.

```
# Estimate transmission distance for simulated data
set.seed(123)
```

```
dist.func <- alist(n=1, a=1/100, rexp(n, a)) # Dispersal kernel
sim <- sim.epidemic(R=2, gen.t.mean=7, gen.t.sd=2, min.cases=100,
                    tot.generations=8, trans.kern.func=dist.func)

# Estimate mean transmission distance
sim.transdist <- est.transdist(epi.data=sim, gen.t.mean=7, gen.t.sd=2, t1=0,
                               max.sep=1e10, max.dist=1e10, n.transtree.reps=10)

sim.transdist
$mu.est
[1] 92.79699
$sigma.est
[1] 91.45614
$bound.mu.est
[1] 131.2348
$bound.sigma.est
[1] 129.3385

# Estimate confidence intervals around mean
sim.transdist.ci <- est.transdist.bootstrap.ci(epi.data=sim,
                                               gen.t.mean=7,
                                               gen.t.sd=2,
                                               t1=0,
                                               max.sep=1e10,
                                               max.dist=1e10,
                                               n.transtree.reps=10,
                                               boot.iter=5,
                                               ci.low=0.025,
                                               ci.high=0.975)
sim.transdist.ci
$mu.est
[1] 131.2124
$mu.ci.low
2.5%
128.2505
$mu.ci.high
97.5%
134.3312
```

Change in mean transmission distance over time

An estimate of μ_k over the duration of an epidemic is indicative of the overall spatial dependence. However, conditions may change over the course of an epidemic that alter the spatial scale upon which transmission operates. To quantify temporal heterogeneity in the mean transmission distance, we have implemented the `est.transdist.temporal` and `est.transdist.temporal.bootstrap.ci` functions, which estimate the change in $\hat{\mu}_k$ over time and its bootstrapped confidence intervals respectively.

When applying the temporal versions of the `est.transdist` functions, it is important to consider the sample size at each time step because the `est.transdist.temporal` function estimates μ_k for all cases leading up to each unique time step. Some time steps at the beginning of an epidemic may be returned as `NA` if there are not enough unique cases to estimate μ_k. Furthermore, in scenarios where time steps in the beginning of an epidemic have low sample sizes, such as an epidemic with a low R_0, $\hat{\mu}_k$ may be over- or under-estimated and display larger confidence intervals due to sampling error. Therefore, we recommend either setting the `t1` argument to the first time step that contains a sufficient sample size, or plotting results along with cumulative sample size as we have done in Figure 3.

```
# Estimate temporal transmission distance for simulated data
set.seed(123)
dist.func <- alist(n=1, a=1/100, rexp(n, a)) # Dispersal kernel
sim <- sim.epidemic(R=2, gen.t.mean=7, gen.t.sd=2, min.cases=100,
                    tot.generations=8, trans.kern.func=dist.func)

# Estimate mean and confidence intervals at each time step
sim.temp.transdist.ci <- est.transdist.temporal.bootstrap.ci(epi.data=sim,
```

```
                                                gen.t.mean=7,
                                                gen.t.sd=2,
                                                t1=0,
                                                max.sep=1e10,
                                                max.dist=1e10,
                                                n.transtree.reps=10,
                                                boot.iter=5,
                                                ci.low=0.025,
                                                ci.high=0.975)

head(sim.temp.transdist.ci)
   t    pt.est    ci.low   ci.high n
1  0        NA        NA        NA 1
2  3        NA        NA        NA 2
3  8        NA        NA        NA 3
4  9  44.61359  35.52047  52.24099 4
5 10 101.55313  43.99469 203.11247 5
6 11 189.29767 113.79560 224.79960 6
```

Application to foot-and-mouth disease

To provide an example of how the functions shown above can be applied to real data, we estimate the mean transmission distance for the 2001 foot-and-mouth epidemic among farms in Cumbria, UK. These data can be found in the `fmd` data object included in the `sparr` package (Davies et al., 2011). It contains transformed (x,y) coordinates of the infected farms and the time step (t) in which it was infected, which is given in days since the index farm was infected (Figure 2). The generation time for foot-and-mouth disease is estimated to have a mean of 6.1 days and a standard deviation of 4.6 days (Haydon et al., 2003), so we use these in the `gen.t.mean` and `gen.t.sd` arguments in the `est.transdist.temporal.bootstrap.ci` function.

```
library(sparr)
data(fmd)
fmd <- cbind(fmd$cases$x, fmd$cases$y, fmd$cases$marks)

head(fmd, n=3)
[,1]     [,2] [,3]
[1,] 333.0328 541.3405   52
[2,] 336.1428 543.3462   46
[3,] 341.4762 551.1794   38

sim.plot(fmd)
```

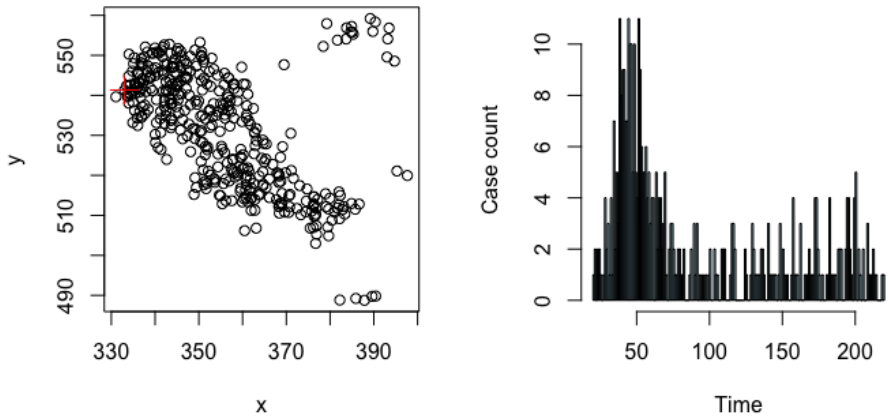

Figure 2: The spatial and temporal distribution of case farms from the 2001 foot-and-mouth epidemic among farms in Cumbria, UK; plotted using the `sim.plot` function. The `x` and `y` axis in the left plot represent transformations of UTM coordinates in kilometers. On the right, case counts are plotted by days since the index case. Data are provided by the `sparr` package (Davies et al., 2011).

```
# NOTE: this function may take a while depending on the data set
fmd.trans <- est.transdist.temporal.bootstrap.ci(epi.data=fmd,
                                                 gen.t.mean=6.1,
                                                 gen.t.sd=4.6,
                                                 t1=0,
                                                 max.sep=1e10,
                                                 max.dist=1e10,
                                                 n.transtree.reps=5,
                                                 boot.iter=10,
                                                 ci.low=0.025,
                                                 ci.high=0.975,
                                                 parallel=TRUE,
                                                 n.cores=detectCores())
par(mfrow=c(1,1))
fmd.trans[,2:4] <- fmd.trans[,2:4]/1000 # Convert to km
plot(fmd.trans$t, fmd.trans$pt.est, pch=19, col='grey', ylim=range(fmd.trans[,3:4], na.rm=T),
     xlab='Time step', ylab='Estimated mean of transmission kernel (km)')

tmp <- seq(1, nrow(fmd.trans), 5)
axis(3, tmp, fmd.trans[tmp,5])
mtext('Sample size (n)', side=3, line=3)

tmp <- which(fmd.trans$n >= 30)[1]
abline(v=tmp, lty=2)
text(16, 1, 'n = 30')

tmp <- tmp:nrow(fmd.trans)
lty <- c(NA,1,2,2)

for(i in 2:4) {
     low <- loess(fmd.trans[tmp,i] ~ as.vector(tmp), span=0.3)
     low <- predict(low, newdata=data.frame(as.vector(tmp)))
     lines(c(rep(NA, tmp[1]), low), lwd=2, lty=lty[i], col='blue')
}
```

Using our approach described above, we estimated the mean transmission distance between case farms in the `sparr` package foot-and-mouth disease data to be $\hat{\mu}_k = 5.8$ km (95% CI = 5.7–6.1 km). Interestingly, this estimate of μ_k is lower than that reported in Salje et al. (2016b), where $\hat{\mu}_k = 9.1$ km (95% CI = 8.4–9.7 km). The difference in $\hat{\mu}_k$ is likely due to differences in data sources. The values estimated in Salje et al. (2016b) include case farms from both Cumbria and Dumbfriesshire, UK with the additional constrain that only case farms where the source farm was confirmed were included. The `sparr` data set, on the other hand, contains all case farms from only Cumbria.

Global clustering: the τ-statistic

Estimating the mean of the spatial transmission kernel (above) provides information on the small spatial scale of individual transmission events. After subsequent generations of transmission where different transmission chains overlap in space, a larger area of elevated disease prevalence will be observed. To describe this larger-scale process, we introduced the τ-statistic in Lessler et al. (2016). The τ-statistic measures global clustering with an epidemiological interpretation—the relative risk of an individual being a related case (under some definition) given they are within a particular distance from another case. The spatial distances where the relative risk is high represent an area of elevated prevalence that is likely to have greater public health utility compared with the scale of individual transmission because interventions must account for ongoing transmission at the population level to contain an outbreak. Therefore, the τ-statistic provides a more intuitive global measure of spatial clustering, which can be interpreted as the relative risk of infection.

Formulation of the τ-statistic has a mathematical relationship to the K-function and pair correlation function. The K-function quantifies the expected number of neighboring points within distance d of a typical point $\mathbf{Z}$ relative to the intensity of the underlying population distribution λ.

$$K(d) = \frac{1}{\lambda} E[\text{neighbors within distance } d \mid x, y \text{ coordinates of } \mathbf{Z}]$$

In the simplest scenario, λ is assumed to be homogeneously distributed, so that $\lambda = N/A$, where N

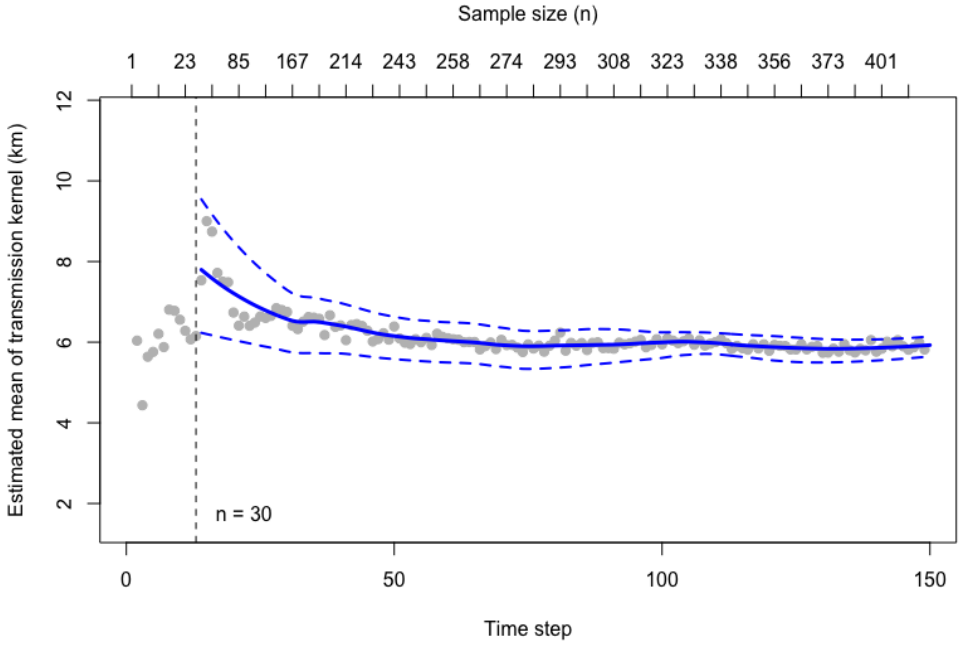

Figure 3: Output from the `est.transdist.temporal.bootstrap.ci` function showing the change in the mean transmission distance over the course of the 2001 foot-and-mouth disease epidemic for case farms in the `fmd` data set in the `sparr` package. The point estimates are plotted as grey circles and a loess smooth of the mean estimate is plotted (blue line) along with its 95% bootstrapped confidence intervals (dashed blue lines). The loess smooth begins with the first time step that contains a cumulative sample size of 30, indicated by the dashed line.

is the total number of cases and A is the total study area. Under the assumption of a heterogeneous underlying population distribution, λ becomes the location specific intensity $\lambda(S)$, where $S \subset A$ within distance d of location $\mathbf{Z}$. In both cases, the K-function is plotted with the theoretical value of the K-function for a homogeneous Poisson process πd^2, which indicates clustering or dispersion relative to complete randomness. The cumulative aspect of the K-function (using all neighbors within distance d) is, however, a well-known constraint that makes it difficult to interpret changes in clustering over distance. The pair correlation function alleviates this constraint by applying the K-function within a distance range (d_1, d_2) and standardizing it by the complete spatial randomness expectation for a homogeneous Poisson process within this range:

$$G(d_1, d_2) = \frac{K(d_1 + \Delta d) - K(d_1)}{2\pi d_1 \Delta d + \pi \Delta d^2},$$

where, $\Delta d = d_2 - d_1$. Both the K-function and pair correlation functions have seen general application due to developments that accommodate inhomogeneous underlying population distribution, clustering between typed points, and edge effects. However, these functions assume complete knowledge of the underlying population distribution and use a null statistical hypothesis of complete spatial randomness, which is precarious for scenarios in epidemiology where the underlying population is unknown and relative risk is used to understand disease dynamics.

To incorporate other metrics of global clustering, the **IDSpatialStats** package provides wrapper functions for calculating both the cross K- and cross pair correlation functions using the `Kcross` and `PCFcross` functions from the **spatstat** package (Baddeley et al., 2016). These wrapper functions allow for straightforward calculation of these statistics using typed epidemiological data that is formatted for **IDSpatialStats** functions (Figure 4).

```
# Calculate cross-K and cross pair correlation functions with simulated data
data(DengueSimRepresentative)

r.vals <- seq(0, 1000, 20)
labs <- seq(0, 1000, 200)

k <- get.cross.K(epi.data=DengueSimRepresentative, type=5, hom=1, het=NULL,
                 r=r.vals, correction='border')
```

```
head(k, n=3)
   r      theo    border
1  0     0.000     0.000
2 20  1256.637  2166.362
3 40  5026.548  5956.887

g <- get.cross.PCF(epi.data=DengueSimRepresentative, type=5, hom=1, het=NULL,
                   r=r.vals, correction='border')
head(g, n=3)
   r theo      pcf
1  0    1 1.000000
2 20    1 1.720848
3 40    1 1.178406

par(mfrow=c(1,2))
plot(k[,3], type='l', lwd=2, xaxt='n', xlab='distance (m)', ylab='cross K function')
lines(k[,2], col='red', lty=2, lwd=2)
axis(1, at=which(r.vals %in% labs), labels=labs)

legend(-3, 4.15e6, legend=c("Theoretical Poisson process", "Observed function"),
       col=c("red", "black"), lty=2:1, box.lty=0, bg='transparent',
       x.intersp=0.7, y.intersp = 1.2)

plot(g$pcf, type='l', lwd=2, xaxt='n', xlab='distance (m)',
     ylab='cross pair correlation function')
abline(h=1, col='red', lty=2, lwd=2)
axis(1, at=which(r.vals %in% labs), labels=labs)
```

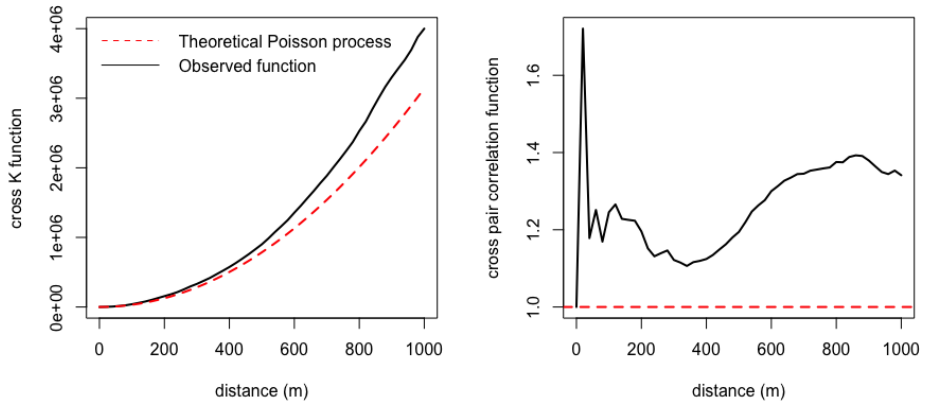

Figure 4: Output from the cross K function (left) and the cross pair correlation function (right) with the observed function shown in black and the value of a theoretical homogeneous Poisson process shown in red.

A measure of relative risk that does not assume knowledge of the underlying population distribution was developed for point pattern data in veterinary epidemiology (Diggle et al., 2005). This function, which is implemented in the `sparr` (Davies et al., 2011) and `spatstat` (Baddeley et al., 2016) packages, uses spatial kernel functions to calculate a ratio of spatial intensity for two different point types $\rho(S) = \lambda_1(S)/\lambda_0(S)$. This formulation can quantify the relative risk for case-control point data or case occurrences with multiple types, but it is not a global clustering statistic.

The τ-statistic can be computed in two ways depending on the underlying assumption that the true population distribution is known. If this assumption is true, then the τ-statistic is similar to other common measures of global clustering that rely on knowledge of the background population distribution to quantify generic clustering of a point process.

$$\hat{\tau}(d_1, d_2) = \frac{\hat{\pi}(d_1, d_2)}{\hat{\pi}(0, \infty)},$$

where $\hat{\pi}(d_1, d_2)$ represents the incidence rate within distance d_1 to d_2 of a case and $\hat{\pi}(0, \infty)$ represents the incidence rate over the entire extent of the study area. This can be implemented by defining the numerator and denominator using the `get.pi` function or by using the generalized `get.tau`

function with `comparison.type = 'representative'` argument. The $\hat{\pi}(\cdot)$ terms in the numerator and denominator use the occurrence data to calculate the incidence rates for linked cases within some defined distance. Therefore, a critical step in performing an analysis with the τ-statistic is specifying which cases are linked through some defined relationship (homologous) and those that are not (non-homologous). Homology can be defined statically or dynamically. When defined statically, the `get.pi.typed` and `get.tau.typed` functions can be used to assign case types based on a `type` column supplied by the data matrix. When defined dynamically, an indicator function $I(\cdot)$ is used to delineate linked and unlinked cases in the data, which allows greater flexibility when defining case type homology.

```
# Calculate tau-statistic using get.pi.typed functions
data(DengueSimRepresentative)

type <- 2 - (DengueSimRepresentative[,'serotype'] == 1)
typed.data <- cbind(DengueSimRepresentative, type=type)
d2 <- seq(20, 1000, 20)
d1 <- d2 - 20

# Static definition of case type homology
num <- get.pi.typed(typed.data, typeA=1, typeB=2, r.low=d1, r=d2)
den <- get.pi.typed(typed.data, typeA=1, typeB=2, r.low=0, r=1e10)
head(num$pi/den$pi, n=4)
[1] 0.2641154 0.2104828 0.2451847 0.2487042

tau <- get.tau.typed(typed.data, typeA=1, typeB=2, r.low=d1, r=d2,
                     comparison.type = "representative") # Equivalent
head(tau, n=4)
  r.low  r       tau
1     0 20 0.2641154
2    20 40 0.2104828
3    40 60 0.2451847
4    60 80 0.2487042

# Calculate tau-statistic using dynamic expression indicating serotype homology
ind.func <- function(a, b){
     if (a[5] == 1 & b[5] == 1) {
          x <- 1
     } else {
          x <- 2
     }
     return(x)
}

num <- get.pi(posmat=DengueSimRepresentative, fun=ind.func, r.low=d1, r=d2)
den <- get.pi(posmat=DengueSimRepresentative, fun=ind.func, r.low=0, r=Inf)
head(num$pi/den$pi, n=4)
[1] 5.084735 4.967885 4.605805 4.409876

tau <- get.tau(posmat=DengueSimRepresentative, fun=ind.func, r.low=d1,  r=d2,
               comparison.type="representative") # Equivalent
head(tau, n=4)
  r.low  r      tau
1     0 20 5.084735
2    20 40 4.967885
3    40 60 4.605805
4    60 80 4.409876

plot(tau$r.low+tau$r/2, tau$tau, type='l', lwd=2, col='blue', xlab='distance (m)')
abline(h=1, lty=2, lwd=2, col='red')
abline(v=100)
```

The interpretation of the τ-statistic is analogous to that of the pair-correlation function in two ways. First, the τ-statistic is not compared to the theoretical measure of a random Poisson process because the metric is an incidence rate ratio with epidemiological meaning. Instead, this measure is plotted in comparison to a ratio of 1, indicating no relative difference in disease risk among homologous cases.

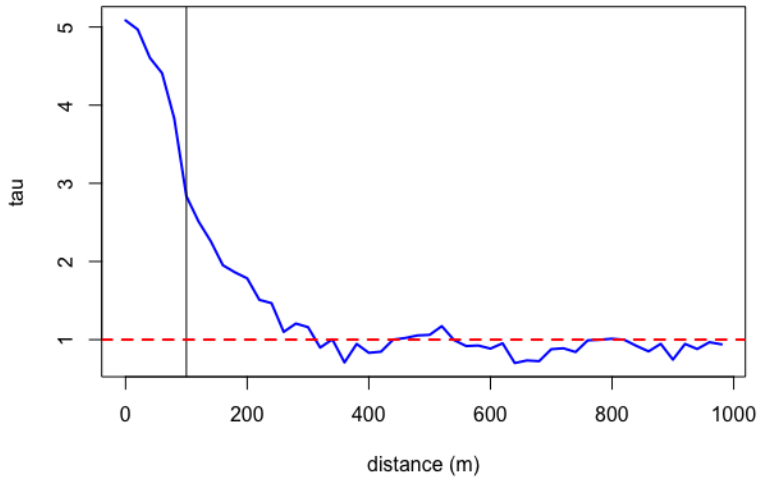

Figure 5: The τ-statistic calculated using the `get.tau` function (blue line) with the theoretical value of no relative difference in disease risk shown by the dashed red line. The vertical black line indicates the mean of the spatial dispersal kernel (100m) used to simulate the `DengueSimRepresentative` data set.

Second, the τ-statistic measures relative risk using case pairs within a distance range $(d_1 \leq d_{ij} < d_2)$. This approach can describe how fine-scale spatial dependence changes over distance. However, if the user wishes to estimate cumulative spatial dependence (analogous to the K-function), then d_1 can be fixed at zero $(0 \leq d_{ij} < d_2)$.

Estimating the τ-statistic with $\hat{\theta}$

If the underlying population distribution is unknown, then the τ-statistic can be computed so that it quantifies global clustering in terms of relative risk. This goes beyond classic measures of clustering by utilizing some relationship between linked and unlinked cases to distinguish transmission chains and quantify relative clustering between them. To do so, we use the function $\hat{\theta}(\cdot)$ which gives the odds ratio of cases related to case i to those independent of i to give an estimate of the τ-statistic that is not biased by assumptions about the underlying population distribution.

$$\hat{\tau}(d_1, d_2) = \frac{\hat{\theta}(d_1, d_2)}{\hat{\theta}(0, \infty)},$$

where,

$$\hat{\theta}(d_1, d_2) = \frac{\sum_{\forall i} \sum_{\forall j} I_1(z_{ij} = 1,\ d_1 \leq d_{ij} < d_2)}{\sum_{\forall i} \sum_{\forall j} I_2(z_{ij} = 0,\ d_1 \leq d_{ij} < d_2)}.$$

The indicator function $I(\cdot)$ is applied to all ij case pairs within distance d_1 and d_2. It returns a binary response which is equal to 1 when case pairs meet user-specified criteria to be homologous and equal to 2 when they are non-homologous. The result is an $i \times j$ relation matrix z_{ij} which is used to find the sums of homologous and non-homologous case pairs. Using an indicator function also allows additional criteria to be used to define case type homology, such as temporal proximity (Figure 6).

```
# Calculate tau-statistic using serotype homology and time
data(DengueSimR01)
d2 <- seq(20, 1000, 20)
d1 <- d2 - 20

# Dynamic expression indicating serotype homology and temporal proximity
ind.func <- function(a, b, t.limit=20){
     if (a[5] == b[5] & (abs(a[3] - b[3]) <= t.limit)){
          x <- 1
     } else {
          x <- 2
     }
     return(x)
}
```

```
num <- get.theta(DengueSimR01, ind.func, r.low=d1, r=d2)
den <- get.theta(DengueSimR01, ind.func, r.low=0, r=Inf)
head(num$theta/den$theta, n=4)
[1] 3.9148969 3.5145802 4.5963608 5.1082210

tau <- get.tau(posmat=DengueSimR01, fun=ind.func, r.low=d1, r=d2,
               comparison.type="independent") # Equivalent
head(tau, n=4)
  r.low  r      tau
1     0 20 3.914897
2    20 40 3.514580
3    40 60 4.596361
4    60 80 5.108221

plot(tau$r, tau$tau, type='l', lwd=2, col='blue', xlab='distance (m)')
abline(h=1, lty=2, lwd=2, col='red')
abline(v=100)
```

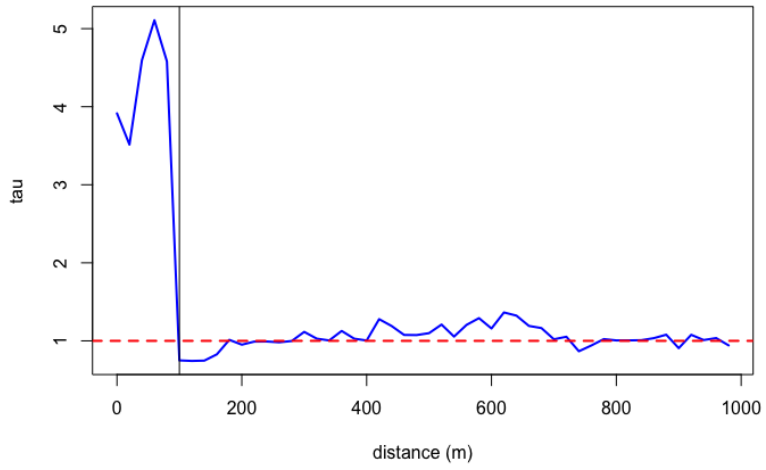

Figure 6: The τ-statistic calculated using `get.tau` with an indicator function based on serotype homology and temporal proximity (blue line) with the theoretical value of no relative difference in disease risk shown by the dashed red line. The vertical black line indicates the mean of the spatial dispersal kernel (100m) used to simulate the `DengueSimR01` data set.

Calculating variance in point estimates

In the examples above, the `get.pi`, `get.theta`, and `get.tau` function families calculate point estimates for $\hat{\pi}$, $\hat{\theta}$, and $\hat{\tau}$ respectively. In scenarios where observation error, sampling bias, or measurement error are expected to introduce additional variance, users may wish to place confidence intervals around these point estimates. For this purpose, each family of functions contains a function ending with a `.bootstrap` suffix, which generates point estimates for `boot.iter` number of bootstrapped samples of the data (Efron and Tibshirani, 1994). Functions ending with a `.ci` suffix are wrappers that calculate user specified confidence intervals based on the bootstrapped samples (Figure 7).

```
# Calculate variance around point estimates of the tau-statistic
data(DengueSimR02)

d2 <- seq(20, 1000, 20)
d1 <- d2 - 20

# Function indicating genotype homology
ind.func <- function(a, b){
     if(a[4] == b[4]){
          x = 1
     } else{
```

```
          x = 2
     }
     return(x)
}

# Bootstrapped estimates of tau
tau.boot <- get.tau.bootstrap(DengueSimR02, ind.func, r.low=d1, r=d2, boot.iter=5)
head(tau.boot, n=4)
  r.low  r       X1       X2       X3       X4       X5
1     0 20 51.04283 49.17736 60.45922 43.36588 37.26332
2    20 40 20.80095 29.62483 26.54935 34.11416 31.32279
3    40 60 34.05415 35.66984 40.21975 31.02943 32.77966
4    60 80 30.52361 35.46972 27.77247 36.64628 32.43156

# Wrapper function of get.tau.bootstrap calculates confidence intervals
tau.ci <- get.tau.ci(DengueSimR02, ind.func, r.low=d1, r=d2, boot.iter=25)
head(tau.ci, n=4)
  r.low  r   pt.est   ci.low  ci.high
1     0 20 44.05161 22.73147 59.44465
2    20 40 30.83943 19.68758 42.05249
3    40 60 37.57434 30.48664 45.78121
4    60 80 33.54134 28.12390 38.76330

plot(tau.ci$r, tau.ci$pt.est, ylim=range(tau.ci[,4:5]), type="l", lwd=2, col='blue',
     xlab='distance (m)', ylab='tau')
lines(tau.ci$r, tau.ci$ci.low, lty=2, lwd=1, col='blue')
lines(tau.ci$r, tau.ci$ci.high, lty=2, lwd=1, col='blue')
abline(h=1, lty=2, lwd=2, col='red')
```

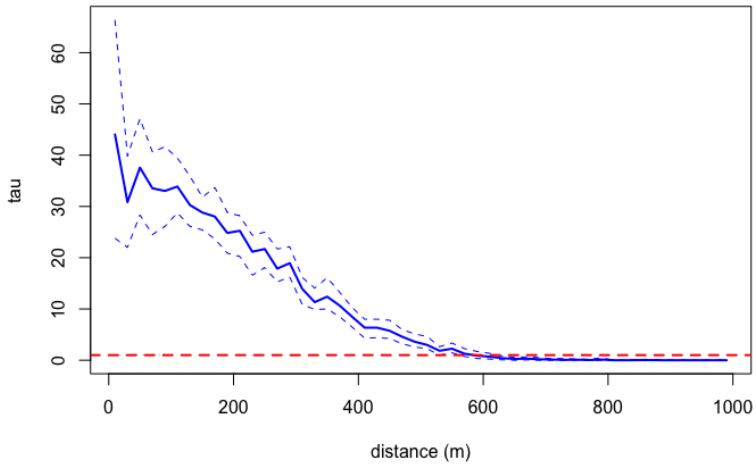

Figure 7: The τ-statistic calculated using `get.tau` with an indicator function based on genotype homology (blue line). The dashed blue lines show the bounds for the 95% confidence intervals calculated by the `get.tau.ci` function. The theoretical value of no relative difference in disease risk shown by the dashed red line.

Null hypothesis testing

A common approach for interpreting spatial clustering statistics includes hypothesis testing using simulation envelopes to assess whether an observed spatial measure is statistically significant (Ripley, 1979; Baddeley et al., 2014). To enable null hypothesis tests, we have implemented a permutation method (Good, 2010) to simulate the nonparametric distribution of $\hat{\pi}$, $\hat{\theta}$, and $\hat{\tau}$ under the null hypothesis of no spatial dependence. The permutation algorithm simulates the null distribution by randomly reassigning case coordinates to observations upon each permutation. Null distributions can be computed using functions ending in the `.permute` suffix and then plotted with observed measures to assess statistical significance as a function of distance (Figure 8).

```
# Compare tau statistic to its null distribution using permutation
data(DengueSimR02)
set.seed(123)

d2 <- seq(20, 1000, 20)
d1 <- d2 - 20

# Compare spatial dependence by time case occurs
type <- 2 - (DengueSimR02[,"time"] < 120)
typed.data <- cbind(DengueSimR02, type=type)

typed.tau <- get.tau.typed(typed.data, typeA=1, typeB=2, r.low=d1, r=d2,
                           comparison.type = "independent")

head(typed.tau, n=4)
  r.low  r       tau
1     0 20 0.4040661
2    20 40 0.5471728
3    40 60 0.7897655
4    60 80 0.8901166

# Perform permutations of observed case times and locations for null distribution
typed.tau.null <- get.tau.typed.permute(typed.data, typeA=1, typeB=2, r.low=d1, r=d2,
                                        permutations=100,
                                        comparison.type = "independent")
head(typed.tau.null[,1:7], n=4)
  r.low  r        X1        X2        X3        X4        X5
1     0 20 1.2570945 0.8530284 3.5019060 1.0326133 1.0775095
2    20 40 1.1448539 0.7045255 0.7224212 2.0742058 0.8754765
3    40 60 0.6947101 0.8904419 0.8249682 0.6990984 0.5128531
4    60 80 1.6266250 1.0916873 0.8326210 0.8432683 1.4815756

# 95% confidence intervals of null distribution
null.ci <- apply(typed.tau.null[,-(1:2)], 1, quantile, probs=c(0.025, 0.975))

plot(typed.tau$r, typed.tau$tau, type='l', lwd=2, ylim=range(c(typed.tau$tau, null.ci)),
     xlab="distance (m)", ylab="tau")
lines(typed.tau$r, null.ci[1,], lty=2)
lines(typed.tau$r, null.ci[2,], lty=2)
abline(h=1, lty=2, lwd=2, col='red')
```

Summary

Conventional spatial statistics are often used to describe the intensity or clustering of point processes. Quantifying spatial dependence of infectious disease spread, however, requires a modified approach that considers overlapping transmission chains and the likelihood of case linkage. Therefore, we have implemented two types of spatial statistics in the **IDSpatialStats** package (the mean transmission distance μ_k, and the τ-statistic) that can be used along with other measures of spatial dependence (e.g. the cross K-function and cross pair correlation function) to understand the spatial spread of infectious diseases.

We showed how to simulate epidemiological data and estimate μ_k, and the τ-statistic, which can be used as templates for other analyses. First, the `transdist` family of functions provides a measure of fine-scale spatial dependence by estimating the mean of the transmission distance $\hat{\mu}_k$ between sequentially linked cases in a transmission chain (Salje et al., 2016b). Second, the `get.tau` family of functions measure spatial dependence on a larger-scale by estimating the τ-statistic, which describes the area of elevated prevalence surrounding cases. This family of functions does so by estimating the relative risk of a case being homologous compared with non-homologous case types. The definition of case type homology is flexible and can utilize temporal or biological information, such as genotype and serotype of the pathogen.

The generalized structure of the `get.tau` family allows for diverse applications of the τ-statistic to epidemiological data. Previous studies have used the τ-statistic to quantify spatial and/or temporal dependence of transmission for Dengue, Cholera, HIV, and Measles disease systems (see Table 2 for

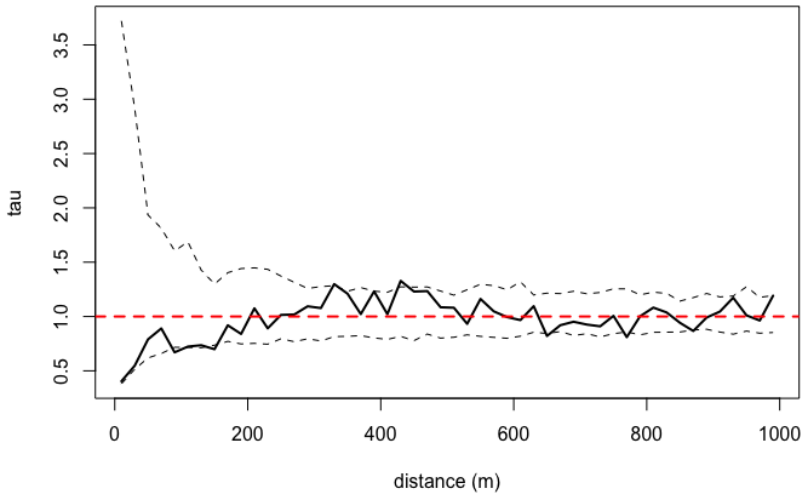

Figure 8: A null hypothesis test for the τ-statistic calculated using the `get.tau.typed.permute` function. The point estimate for $\hat{\tau}$ is shown by the black line and the 95% confidence bounds of permuted $\hat{\tau}$ values are indicated by the dashed lines. The theoretical value of no relative difference in disease risk is shown by the dashed red line. The plot indicates that the point estimates for $\hat{\tau}$ are not statistically significant in this example because they are within the bounds of the null distribution.

detailed descriptions). These studies illustrate that, regardless of the system under study, analyses are enhanced when bootstrapping, permutation tests, and/or assessment of observation error is employed to understand the distribution of error and statistical significance for estimates of the τ-statistic.

The **IDSpatialStats** package is undergoing continued development. Future directions include expanding the implementation of the τ-statistic to facilitate estimation of spatio-temporal dependence by incorporating a temporal interval into the spatial search window. This technique was used by Salje et al. (2012) in the form of the ϕ-statistic to estimate $\hat{\phi}(d_1, d_2, t_1, t_2)$. Additional developments include a theoretical framework for the τ-statistic that incorporates uncertainty due to pathogen generation time, and to define case type homology more continuously using genetic distance matrices. We hope these developments will enable users to address more complex questions and incorporate more sources of uncertainty into estimates of spatial dependence. Check Github at `https://github.com/HopkinsIDD/IDSpatialStats` for latest development release.

Table 2: Descriptions of how previous studies have used the τ-statistic to quantify spatial dependence for infectious diseases. Listed in chronological order.

Description	Citation
Spatial and temporal dependence of homotypic and heterotypic Dengue virus serotypes over a 5 year period in Bangkok, Thailand	Salje et al. (2012)
Clustering of HIV prevalence and incidence around HIV-seropositive individuals using cohort data from rural Rakai District, Uganda	Grabowski et al. (2014)
Overview of the τ-statistic, its performance given observation error, and illustrations using Dengue, HIV, and Measles	Lessler et al. (2016)
Spatial dependence of seroconverted individuals in the 2012–2013 Chikungunya outbreak in the Phillipines	Salje et al. (2016a)
Comparison of spatial dependence in endemic transmission of Dengue virus serotypes in Bangkok and Ho Chi Min City, Thailand	Quoc et al. (2016)
Risk of Cholera transmission within spatial and temporal zones after case presentation during urban epidemics in Chad and D.R. Congo	Azman et al. (2018)
Summary statistic to fit micro-simulations of Cholera interventions to epidemic data using Approximate Bayesian Computation	Finger et al. (2018)
Temporal clustering of subclinical infections and homologous serotypes within schools using Dengue cohort data in Thailand	Salje et al. (2018)