# Accelerating the discovery of space-time patterns of infectious diseases using parallel computing

Alexander Hohl, Eric Delmelle, Wenwu Tang and Irene Casas
Accepted May 17 2016 for Spatial and Spatiotemporal Epidemiology

**Abstract**
Infectious diseases have complex transmission cycles, and effective public health responses require the ability to monitor outbreaks in a timely manner. Space-time statistics facilitate the discovery of disease dynamics including rate of spread and seasonal cyclic patterns, but are computationally demanding, especially for datasets of increasing size, diversity and availability. High-performance computing reduces the effort required to identify these patterns, however heterogeneity in the data must be accounted for. We develop an adaptive space-time domain decomposition approach for parallel computation of the space-time kernel density. We apply our methodology to individual reported dengue cases from 2010 to 2011 in the city of Cali, Colombia. The parallel implementation reaches significant speedup compared to sequential counterparts. Density values are visualized in an interactive 3D environment, which facilitates the identification and communication of uneven space-time distribution of disease events. Our framework has the potential to enhance the timely monitoring of infectious diseases.

Keywords: dengue fever, parallel computing, space-time analysis

# 1. Introduction

*1.1. Detecting Disease Outbreaks*

Infectious diseases have complex transmission cycles, and an effective public health response requires the ability to monitor and analyze outbreaks under critical space-time conditions (Eisen and Eisen 2011). Space-time analytics and geovisualization are particularly attractive to analyze spatial data with a time stamp (Jacquez, Greiling, and Kaufmann 2005; Rogerson and Yamada 2008; Robertson et al. 2010; Kulldorff 2010), as they facilitate the discovery of inherent patterns (rate of disease spread, cyclic pattern, direction, intensity and risk of diffusion to new regions). The identification of a cluster of illness provides critical intelligence for response; timely and focused monitoring is thus a critical element to reduce the burdens associated with diseases. The detection of space-time clusters can be computationally demanding, and this issue is exacerbated with spatiotemporal datasets of increasing size, diversity and availability (Grubesic, Wei, and Murray 2014; Robertson et al. 2010). Accelerated processing capabilities are therefore critical to reduce the computational effort when conducting space-time analysis on epidemiological datasets, and particularly so at the individual level. However, careful spatiotemporal domain decomposition is often necessary to prevent workload imbalances, thereby reducing computing inefficiency. Heterogeneity in the data has to be accounted for when designing new algorithms capable of implementing parallel strategies and integrating time along spatial dimension.

A series of statistical approaches has been used to detect spatial or spatiotemporal clusters of infectious diseases. The Knox test for space-time interaction evaluates the presence of a space-time cluster at given spatial and temporal distances (Kulldorff and Hjalmars 1999). Knox' method is limited due to its arbitrary definition of closeness (Robertson et al. 2010), and the Mantel's test (Mantel 1967) incorporates the notion of distance decay in that nearby pairs of events are more important than distant pairs. The space-time Ripley's K function evaluates the magnitude of space-time clustering at different spatial and temporal scales (Bailey and Gatrell 1995). The spatial scan statistic (Kulldorff 1997) identifies the most likely disease clusters in a study area by maximizing the likelihood that disease cases are located within a set of concentric circles that are moved across the study area. In a space-time context, the scan statistic uses a cylinder instead of a circle, where the vertical axis represents time (Kulldorff et al. 2005).

In addition to scan statistics, autocorrelation-based methods have been widely used for identifying clusters in epidemiology (Mclafferty 2015). Global methods, such as Moran's I (Moran 1950), tell us whether clustering of similar attribute values is present within the study area, while its local version (local indicators of spatial association, LISA) identifies locus and shape of these clusters. Finally, kernel density estimation (KDE) techniques are used to generate continuous surfaces of disease intensity (Carlos et al. 2010; Delmelle, Zhu, et al. 2014). The temporal extension of the KDE is known as the space-time kernel density estimation (STKDE) and essentially maps a volume of disease intensity along the space-time domain (Nakaya and Yano 2010). However, the above methods are computationally intensive (Robertson et al. 2010), especially when Monte Carlo simulations are used for significance testing, and when the temporal dimension is added, resulting in long execution times when compared to their planar counterparts (Costa, Assunção, and Kulldorff 2012). Expedited processing capabilities are critical for analyzing spatiotemporal epidemiological data of increasing size, diversity and availability. High-performance and parallel computing offer the capacity to solve computationally demanding problems in limited time.

### *1.2. Parallel Computing*

Parallel computing is based on a divide and conquer strategy that breaks down a problem into sub-problems that are small enough to be handled computationally. Then, the solutions to the sub-problems are aggregated to form a solution for the original problem (Wilkinson and Allen 2004). The general approach for parallel computing is to decompose a dataset into smaller subsets, for example, along the spatial or temporal dimension, distribute the resulting subdomains to multiple processors for parallel processing, and finally collect and reassemble the results (Ding and Densham 1996). However, to prevent workload imbalance among processors and, therefore, processing inefficiency, the spatially explicit characteristics of the data often need to be accounted for (Wang and Armstrong 2003). While random or uniform data can be decomposed by non-adaptive regular tessellations, doing so for clustered datasets results in heterogeneous subdomains as they contain uneven quantities of data (Ding and Densham 1996). Such subdomains exhibit variation in computational intensities (Wang and Armstrong 2009) resulting in workload imbalance and inefficiency. Recursive domain decomposition methods, such as quadtrees, have been widely used for mitigating workload imbalance, especially for spatially heterogeneous data (Turton 2003; Wang and Armstrong 2003).

Despite the recent popularity of incorporating time dimension in geographic models (Kwan and Neutens, 2014), the recursive decomposition of spatiotemporal datasets has been insufficiently addressed in the literature. Most of the work reported is based on static spatial domain decomposition. For example, spatial domain decomposition has been used for parallel computation of the $G_i^*(d)$ statistic (Armstrong and Marciano 1995; Wang, Cowles, and Armstrong 2008). Liu and colleagues compared non-adaptive with adaptive domain decomposition for parallel processing, and found that adaptive decomposition often leads to increase in workload balance (Liu et al. 2010). Spatial domain decomposition, including both regular (block) and ordered (cyclic; taking into account heterogeneity in spatial data) strategies, has been applied to parallelize the AMOEBA algorithm for detection of spatial clustering patterns (Widener, Crago, and Aldstadt 2012). Static spatial domain decomposition has been used to compute space-time kernel density (STKDE) in parallel on reported dengue fever cases (Delmelle, Dony, et al. 2014). The latest efforts of accelerating clustering algorithms include the use of general-purpose graphics processing units (GPGPU), for instance, to select optimal bandwidths for kernel density estimation (Andrzejewski, Gramacki, and Gramacki 2013), and to compute the Ripley's K function on massive spatial point data (Tang, Feng, and Jia 2014).

### *1.3. Objectives*

In this article, we develop a parallel computing approach based on adaptive space-time domain decomposition for the acceleration of the space-time kernel density estimation (STKDE), a computationally demanding space-time statistic. We apply our methodology on an epidemiological dataset of reported dengue fever cases in the city of Cali, Colombia for the years 2010 and 2011. We implement a parallel computing approach to conduct the space-time K function test, on both observed cases and population adjusted Monte Carlo simulations. Optimal space-time K-function parameters serve as inputs for the STKDE. Based on the K-function analysis, we choose the spatial and temporal bandwidths for STKDE.

# 2. Materials and Methods

*2.1 Data and Study Area*

The city of Santiago de Cali (from here on referred to as Cali) is located in the valley of the Cauca River; limited to the east by the river and to the west by a mountain system. It has a tropical climate with two distinct rainy seasons, from April to July and September to December. The yearly average temperature is 26°C (79°F) which results in perfect conditions for the Aedes Aegypti mosquito to reproduce (de Cali 2008). The city of Cali with a population of around 2.5 million people is considered an endemic zone for dengue fever. Similarly to other urban areas in the developing world, Cali is characterized by unplanned urbanization (Restrepo 2011). Multiple neighborhoods along the river banks to the east of the city are the result of squatter settlements. A similar phenomenon takes place in the western foothills. Based on data from the health municipality of the city of Cali (Cali 2010), until 2009 three severe dengue outbreaks occurred: 1995, 2002, and 2005. However, two more outbreaks occurred in both 2010 and 2013.

The dengue fever dataset (Figure 1) corresponds to cases reported to the "Sistema de Vigilancia en Salud Pública" (SIVIGILA, English: Public health surveillance system) for the city of Cali during 2010 and 2011. The system is updated on a daily basis with records of individuals that have been diagnosed with dengue fever. The record includes, among others, individual information about the patient (e.g. gender, home address and education among others), date of diagnosis, symptoms, and final condition. Addresses are standardized to a common address format, and spelling and other syntactical errors are manually corrected (Delmelle et al. 2013). Using the home address of each patient, the data is geocoded and masked to the nearest street intersection level to maintain privacy (Kwan, Casas, and Schmitz 2004).

In 2010, 9606 were successfully geocoded (11760 reported cases, or 81.7%), whereas in 2011, 1562 cases were successfully geocoded (1822 reported cases, or 85.7%). The probability of a dengue fever case to be successfully reported and geocoded is rather difficult to estimate. Access to healthcare facilities for diagnosis is not an issue, as the diagnosis can occur close to home in nearby health centers and posts. The geocoding process is more likely to introduce bias due to inexistent or poorly documented road infrastructure in squatter settlements, and because some individuals (1) choose not to report their home address in order to maintain privacy or (2) may only remember an incomplete address that cannot be geocoded.

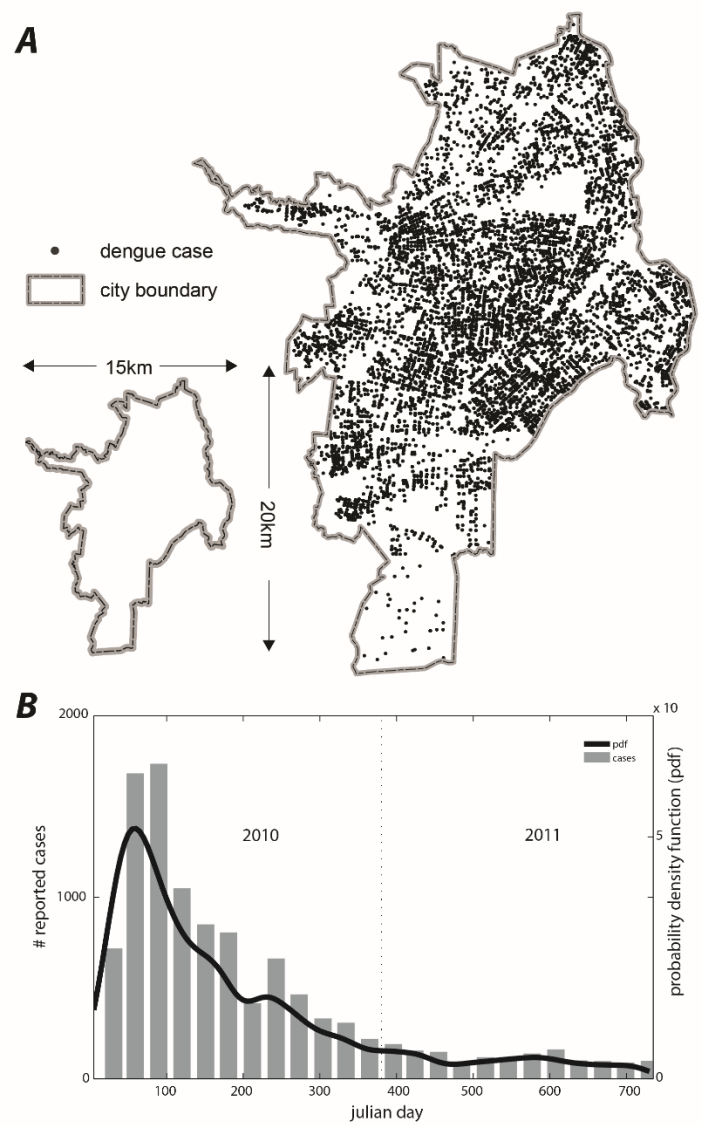



Figure 1 Spatial distribution of geocoded dengue cases from 2010 to 2011 in *A* and their temporal distribution in *B*. Note that each bar in *B* represents one month of the year.

*2.2 Space-Time Ripley's K Function*

To estimate space-time clustering among reported dengue fever cases, we follow a two-step approach (see Figure 2 for framework design). First, we conduct a space-time K function analysis to evaluate the magnitude of spatial and temporal bandwidths at different scales. We do so by comparing the K function values of both observed and population adjusted simulated datasets, computed in parallel. Second, we use these bandwidths for spatiotemporal domain decomposition, which creates a collection of subdomains that are distributed to multiple CPUs (Central Processing Units) for parallel STKDE.

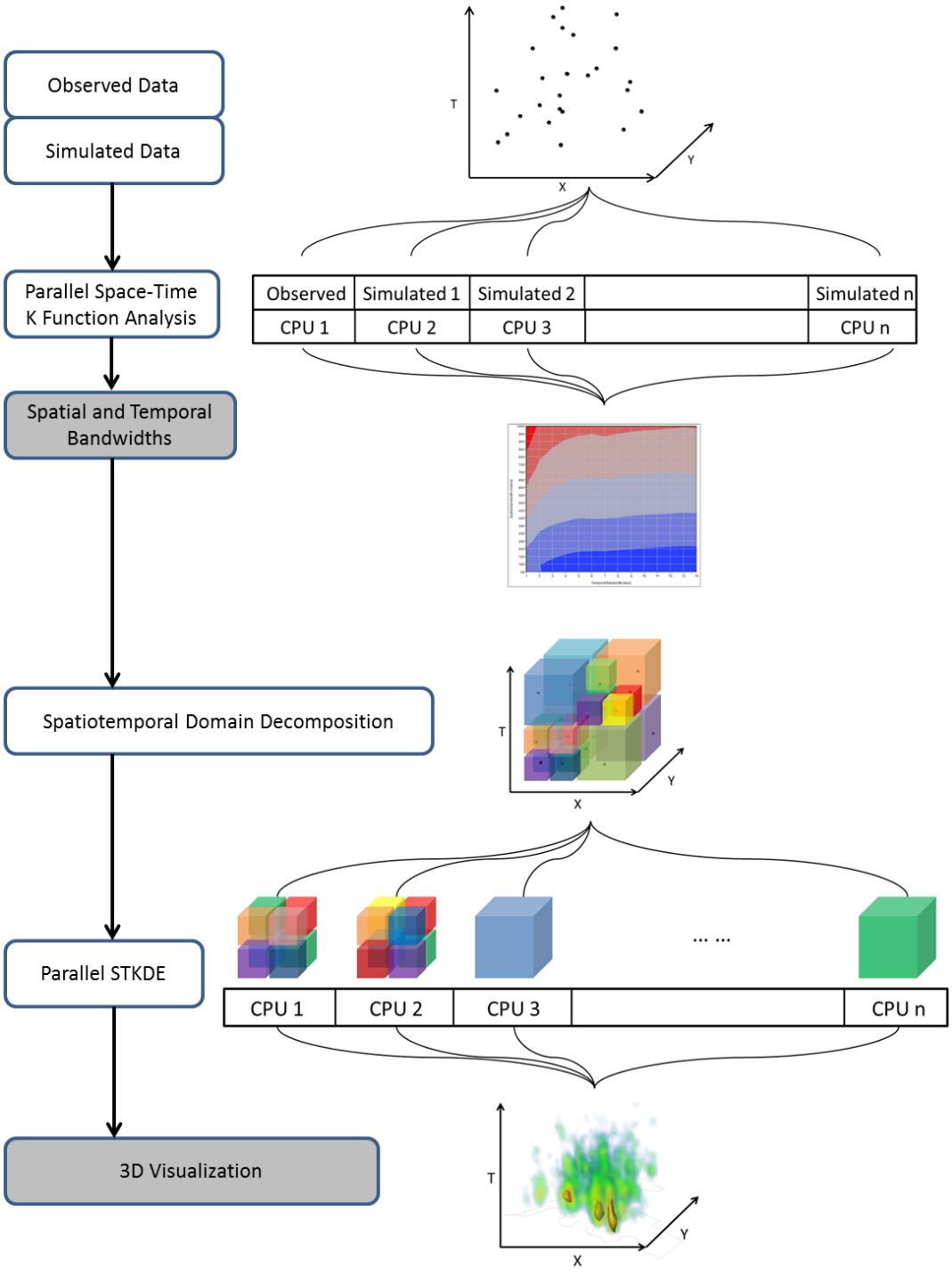



Figure 2: Framework of parallel space-time analysis of Dengue fever cases (STKDE: space-time kernel density estimation; CPU: central processing unit).

The space-time Ripley's K function evaluates the magnitude of space-time clustering (random, clustered, or regular) along multiple spatial and temporal scales (Bailey and Gatrell 1995). It takes into account both the number of points and the distance between them for the

quantification of spatiotemporal clustering. It is based on the second-order property (i.e. variance) of point events (Dixon 2002) and allows for comparison of point patterns and their spatiotemporal structures over multiple scales. Given a specific set of spatial and temporal distances, the space-time Ripley's K function is the expected number of point events within the distance ranges divided by the intensity (first-order property) of the point pattern:

$$K(d,t) = \frac{E(d,t)}{\lambda} \quad (1)$$

where $E(d,t)$ is the expected number of point events within spatial and temporal lag $d$ and $t$. The spatial and temporal lags form a cylinder with radius $d$ and height $t$, which is centered on each point in the study area/period. $\lambda$ can be estimated as $n/(A*T)$, where $n$ is the number of points, and $A$ is the area of the study region, and $T$ the study period. $A*T$ amounts to the volume of the irregular prism that has the study region as its base and the temporal dimension (study period) as its height. The space-time Ripley's K function represents a cumulative distribution of point events over distance and time. If a point pattern follows the property of complete spatiotemporal randomness (CSTR), we expect $K(d,t) = \pi d^2 t$, where $\pi d^2 t$ is the volume of the cylinder defined by $d$ and $t$. For a clustered pattern within spatial and temporal lags $d$ and $t$, $K(d,t) > \pi d^2 t$; otherwise, $K(d,t) < \pi d^2 t$ for a regular pattern. The K function can be estimated by the following formula:

$$\widehat{K}(d,t) = \frac{A*T}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{I_{ij}(d,t)}{w_{ij}} \quad (2)$$

where $I_{ij}(d,t)$ is a binary function that indicates whether a point $j$ is located within $d$ and $t$ from point $i$. $I_{ij}(d,t) = 1$, if $d_{ij}$ and $t_{ij}$ (the spatial and temporal distance between point $i$ and $j$) are shorter or equivalent to $d$ and $t$; otherwise, $I_{ij}(d,t) = 0$. $w_{ij}$ is a weight function that corrects edge effects because cylinders used for counting points within $d$ and $t$ might intersect the boundary of the study region/period and therefore, introduce bias. For a thorough discussion of correction methods for dealing with edge effects, see (Gabriel 2014; Yamada and Rogerson 2003). In this study, we used the isotropic edge correction method, which is the spatiotemporal equivalent of Ripley's circumference method (Ripley 1977).

The structural characteristics of a spatiotemporal point pattern can be evaluated by comparing $\widehat{K}(d,t)$ with the volume of the cylinder of radius $d$ and height $t$, (i.e. $\pi d^2 t$). The K function can be transformed to $L(d,t)$, obtaining a benchmark of zero, which allows for direct comparison of $L$ across all $d$ and $t$ assessed, using Equation 3:

$$\hat{L}(d,t) = \sqrt{\frac{\widehat{K}(d,t)}{\pi t}} - d \quad (3)$$

Therefore, a spatiotemporal point pattern within distances $d$ and $t$ is clustered if $\hat{L}(d,t) > 0$. If $\hat{L}(d,t) < 0$, then the spatiotemporal point pattern is regular within $d$ and $t$. If $\hat{L}(d,t) = 0$, then the pattern conforms to CSTR.

We statistically assessed the scale of clustering by finding the spatial and temporal scales at which the difference between L-values of the observed cases and those of the upper simulation envelope was maximal. Upper and lower envelopes were generated by taking the maximum and minimum L-value among 100 population adjusted random datasets for each combination of

spatial and temporal bandwidth. We simulated dengue cases within each barrio (smallest census level at which population is reported, *n*=340 in Cali), according to 2010 population figures. Specifically, if a barrio contained 10% of the population, we randomly generated 10% of dengue cases within the boundary of the barrio. The temporal signature of each case was chosen between the first diagnosed and reported case on January 3rd, 2010, and the last case on December 31, 2011. We parallelized the Monte Carlo runs of the space-time Ripley's K function: each run was deployed on an individual processor (the so-called embarrassingly parallel computing; see Wilkinson and Allen 2004).

The observed point pattern is spatially and temporally inhomogeneous, as the pattern of disease incidence reflects both, the density of the population at risk and systematic temporal variation due to seasonality and progression of the disease. Approaches to deal with the inhomogeneity include decomposition of the study area/period into sub-regions/periods, for which the point process is homogeneous (Bolibok 2008). Alternatively, the spatiotemporal inhomogeneous K-function (STIK-function) can be used to analyze second-order properties, which necessitates estimating first-order intensity, assuming the separability into its spatial- and temporal components. Kernel estimation is required for estimating the spatial intensity, where the bandwidth is chosen to minimize the estimated mean square error, whereas the temporal intensity stipulates a log-linear regression model (Gabriel and Diggle 2009). We intend to incorporate the STIK-function in future work.

*2.3 Space-Time Kernel Density Estimation*

In order to detect spatiotemporal patterns in our data, we use the space-time kernel density, which is a temporal extension of the traditional kernel density estimation (KDE) used for identifying spatiotemporal patterns of underlying datasets. The density estimates are visualized within the space-time cube framework using two spatial (*x*, *y*) and a temporal dimension (*t*) (Delmelle, Dony, et al. 2014; Demšar and Virrantaus 2010; Nakaya and Yano 2010).

The output is a 3D raster volume where each voxel (volumetric pixel) is assigned a density estimate based on the surrounding point data. The space-time density is estimated by Equation 4 (same notation as Delmelle et al., 2014):

$$\hat{f}(x,y,t) = \frac{1}{nh_s^2h_t}\sum_i i(d_i < h_s, t_i < h_t) k_s\left(\frac{x-x_i}{h_s}, \frac{y-y_i}{h_s}\right) k_t\left(\frac{t-t_i}{h_t}\right) \quad (4)$$

Density $\hat{f}(x,y,t)$ of each voxel *s* with coordinates (*x*, *y*, *t*) is estimated based on neighboring data points ($x_i$, $y_i$, $t_i$). Each point located within the neighborhood of the voxel is weighted using the spatial and temporal Epanechnikov kernel functions, $k_s$ and $k_t$, respectively (the closer the data point, the higher the weight)(Epanechnikov 1969). The spatial and temporal distances between voxel and data point are given by $d_i$ and $t_i$ respectively. If $d_i$ and $t_i$ are smaller than the spatial ($h_s$) and temporal bandwidths ($h_t$) respectively, the indicator function $I(d_i < h_s; t_i < h_t)$ equals 1, otherwise 0. For visualization of STKDE, we set $h_s$ and $h_t$ to 500 meters and 7 days, which was determined preliminary analysis of the space-time K function. We used a spatiotemporal voxel-resolution of 100m * 100m * 1 day within our experimental treatments. STKDE values are standardized and adjusted for 2010 population density values, which are also standardized.

The sequential STKDE algorithm has an algorithmic complexity of *V***N*, where *V* is the number of voxels and *N* is the number of data points. It is implemented as a nested loop structure where the outer loop iterates through the voxels and the inner loop through the data points,

calculating the space-time distances for each voxel/point pair and comparing them to the bandwidths (Figure 3).

```
ALGORITHM STKDE(xyzList, hs, ht)
BEGIN ALGORITHM
for (xC=xmin;xC<=xmax;xC+xRes):          # for all x-coordinates
    for (yC=xmin;yC<=ymax;yC+yRes):          # for all y-coordinates
        for (zC=xmin;zC<=zmax;zC+zRes):          # for all  t-coordinates
            density = 0.0
            for xD, yD, zD in xyzList:                    #loop through disease cases
              if hs >= (xD - xC)^2 + (yD - yC)^2:          #if within spatial bandwidth
                        if ht >= |zD – zC|:                    #if within temporal bandwidth
                              u = (xD-xC) / hs
                              v = (yD-yC) / hs
                              w = (zD-zC) / ht
                             density += 0.5 * π * (1 – u^2 – v^2) * 0.75 * (1 – w^2)
END ALGORITHM
```

Figure 3: Sequential STKDE algorithm.

*2.4 Spatiotemporal Domain Decomposition and Parallel Computing*

To conduct the parallel STKDE, we implemented the divide and conquer strategy by recursively decomposing the dengue fever dataset for subsequent distribution of the resulting subdomains to processor queues for concurrent processing (Figure 2). Following the methodology described in (Hohl, Delmelle, and Tang 2015), we created subdomains of similar computational intensity in order to achieve equal workloads among CPUs. In order to account for heterogeneity in the dengue fever dataset, we used recursive spatiotemporal domain decomposition, which is a method where the solution to a problem depends on solutions to smaller instances of the same problem (Graham 1994).

The decomposition algorithm proceeds as follows (Figure 2). The algorithm starts by defining the bounding box of the dataset, using the minimum and maximum values of each dimension and generates 8 cuboid subdomains by dividing each of the three axes into two equal parts. The algorithm then iterates through each subdomain and keeps decomposing recursively until no subdomain contains more points than the specified threshold (50 in our paper), as long as the ratio between the subdomain volume and the combined subdomain and buffer volume stays above 0.1 (to prevent unnecessarily small subdomains). Subdomains containing no points are discarded from subsequent processing, which reduces the computational workload and execution time regardless of whether parallel computing is used or not. To avoid edge effects in the STKDE -which are likely to occur near subdomain boundaries due to the spatial ($h_s$) and temporal bandwidth ($h_t$)-, we implemented space-time buffers of distance $h_s$ and $h_t$ around all subdomains. The number of subdomains resulting from decomposition depends on the choice of bandwidths. When a data point falls outside a subdomain but inside its buffer, the point is still assigned to that subdomain and hence, contributes towards the number of points for comparison against the threshold. Buffers from neighboring subdomains overlap with each other, and data points that fall within these areas are assigned to both subdomains. Therefore, a data point can be assigned to a maximum of 8 subdomains, which creates considerable data redundancy. To ensure workload balance, we define the target load as the cost of the entire computation divided by the

number of processors. For each processor, we keep adding from the sequence of subdomains obtained from 3D to 1D mapping by space-filling curve (Bader 2012), until the target load is reached. Although the number of assigned subdomains may vary among processors, the workload remains similar.

We use a high-performance computing cluster with 32 nodes connected through an infiniband network switch (Pfister 2001). Each computing node has 12 CPUs and 12GBs of memory, in total 384 CPUs (Intel Xeon processor with 2.67GHz clock speed) for the computing cluster. We use the statistical software R (package: stpp) to conduct the space-time K function (Gabriel, Rowlingson, and Diggle 2013), while the STKDE algorithm is implemented in the Python environment, and the results visualized in Voxler (Golden Software, Colorado), an interactive 3D modelling environment.

To evaluate the sensitivity of our parallel approach to kernel bandwidth and the number of parallel processors, we conducted multiple experimental treatments using several combinations of spatial and temporal bandwidths, and varying numbers of processors. We chose to use computing time, speedup and efficiency as metrics to evaluate the parallel computing performance of our space-time analysis approach. Both speedup and efficiency are performance metrics based on computing time. Speedup (S) is a metric that is based on the ratio of computing time from a single CPU (sequential) over that from multiple CPUs in parallel (see equation 6) (Wilkinson and Allen 2004). Efficiency (E) is the ratio of speedup value over the number of CPUs (see equation 7).

$$S = \frac{T_s}{T_p} \tag{6}$$

$$E = \frac{S}{N} \tag{7}$$

where $S$ is the speedup of our parallel approach. $T_s$ is the sequential computing time and $T_p$ the parallel computing time. The theoretical maximal value of speedup is equivalent to the number of CPUs used for parallel execution. However, real speedup values are usually lower than the theoretical upper limit. The speedup value of a parallel computing algorithm is positively related to computing performance: the higher the speedup value, the better the computing performance our algorithm has. Speedup and efficiency have been extensively used to evaluate the computing performance of parallel algorithms (Wilkinson and Allen 2004).

# 3. Results and Discussion

*3.1 Space-time K-function*

Figure 3 illustrates the difference in $L$-values (see equation 3) between observed dengue fever cases and upper confidence envelope of the population adjusted simulated datasets for the entire study area/period. The confidence envelope is constructed based on maximum and minimum values of 100 Monte Carlo runs. Clustering of dengue fever cases tends to be strong where the difference between observed and simulated values becomes large. This difference is positive across all bandwidths (Figure 4), suggesting that dengue fever cases exhibit clustering throughout all scales assessed, and the clustering becomes strong for large spatial- and small temporal scales (e.g. $h_s$ = 1000m and $h_t$ = 1day). With decreasing spatial- and increasing temporal bandwidth, the clustering pattern becomes weaker, especially along the spatial

dimension. Based on L function results, we chose the bandwidths $h_s$ = 500m and $h_t$ = 7d for STKDE, which corresponds to medium clustering intensity. The total sequential computing time for Monte Carlo run of space-time Ripley's K function is 322 seconds (about 6 minutes). For the parallel computing, we deployed each Monte Carlo run on a single processor. This leads to 3.24 seconds of parallel computing time for 100 Monte Carlo runs.

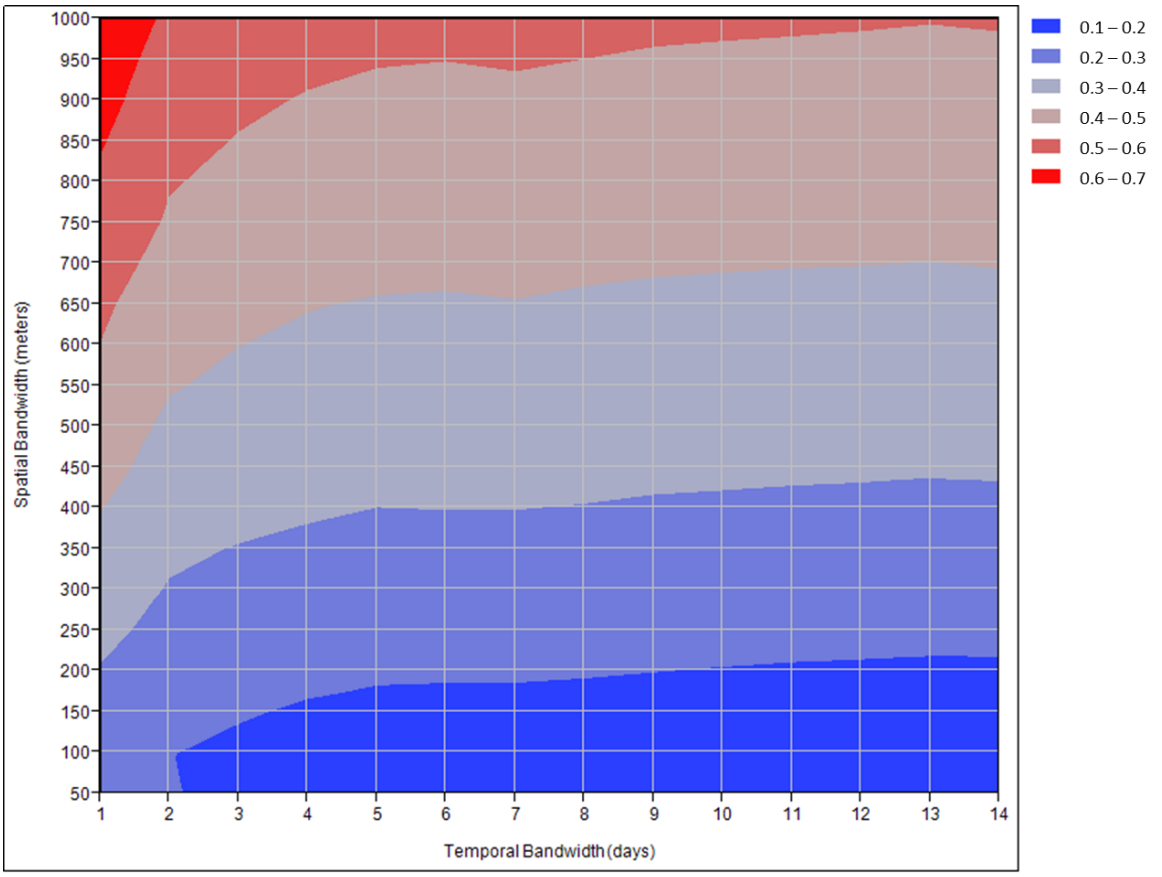


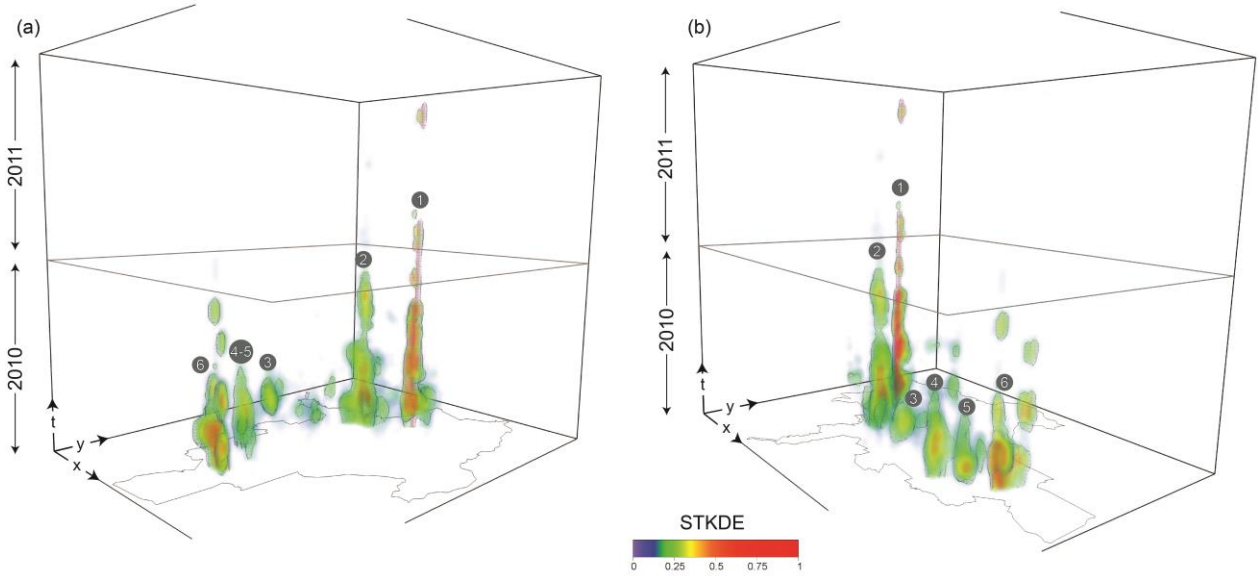

Figure 4: Difference in L function values between observed data and upper envelope of the simulated datasets.

3.2 *Space-Time Kernel Density Visualization*

Space-time kernel density estimates are rendered in Voxler, where each voxel is color-coded according to its population adjusted STKDE value. The level of transparency of each of the voxel can be adjusted to focus on those regions with higher STKDE values. We reinforce clusters of high STKDE values by means of isovolumes which capture the shape of clusters in both space and time. Figure 5 summarizes population adjusted STKDE values using a spatial bandwidth $h_s$=500m and $h_t$=7days, at a discretization level of 100m*100m*1 day, resulting in 16,442,664 voxels.

The space-time distribution of cases is particularly important in the beginning of 2010 and exhibits much lower values in 2011 (Figures 2 and 5). We identify several groups of high population adjusted STKDE values in 2010; such as cluster 1 directly located to the south of Cali River and in the close proximity of *Parque de la Cana*, an attraction park with swimming pools and small ponds (Delmelle et al. 2013). Cluster 2 occurs in the northern part of the city, directly to the South of the Cali River, a river flowing through the city, draining into the Cauca River.

Clusters 1 and 2 have in common that individuals in those areas tend to live in crowded dwellings with poor water supply making them a more vulnerable population (Delmelle et al. 2013). Cluster 3 is located in the old core of the city, but characterized by a relatively low population, which may explain higher population adjusted STKDE values. Clusters (4) and (5) are located on a military base with individuals living in a confined space; the shape of the clusters is characterized by a small spatial extent but large temporal period. Cluster 6




STKDE values for 2011 suggest a decrease of dengue fever cases in the periphery and a higher concentration in the center of the city.

Figure 5: Three dimensional visualization of the population adjusted space-time kernel density estimates (spatial bandwidth: 500m and temporal bandwidth: 7days). Perspective from the Southeast in (a) and from the Southwest in (b).

*3.3 Parallel Computing Performance Analysis*

Figure 6 illustrates the computing performance of our parallel STKDE method, including sequential computing time. Computing time of STKDE algorithm is affected by spatial and temporal bandwidths. Large spatial (or temporal) bandwidths require a longer computing effort. For example, the treatment for 250 meter of spatial bandwidth and 3 days of temporal bandwidth only exhibits a sequential time ($T_s$) of 2,067 seconds (0.59 hours). However, when the spatial bandwidth increases to 2,500 meters and temporal bandwidth to 14 days, computing time becomes 40,297 seconds (about 11.19 hours). The neighborhood search dramatically increases when larger spatiotemporal bandwidths are used, suggesting that the STKDE algorithm is computationally demanding. When we apply parallel computing solutions, STKDE can be accelerated substantially. For example, computing time drops from 11.19 hours down to 244 seconds (about 4 minutes) when 200 CPUs were used for the treatment with 2,500 meter of spatial bandwidth and 14 days of temporal bandwidth. Correspondingly, the speedup for this treatment is 165 (i.e., acceleration for parallel STKDE computation using 200 processors is 165

times) and efficiency is 82.5% (Figure 7). Large spatial/temporal bandwidths tend to have high parallel computing performance. When spatial bandwidths are small (250 m here), parallel computing performance tends to be low when the number of CPUs used is high. This can be attributed to the low number of sub-domains decomposed from the spatiotemporal domain decomposition algorithm (3,420 subdomains decomposed for 250-m and 14-day spatiotemporal bandwidth). Thus, as the number of CPUs becomes large, the number of sub-domains assigned to each processor decreases—leading to a higher likelihood of workload imbalance among processors, because performance increase is limited to the subdomain that constitutes the largest share of the workload, and adding additional processors will not further accelerate the computation. In other words, when we utilize parallel computing to accelerate STKDE, large spatiotemporal bandwidths, which requires more computational support, tend to leverage parallel computing resources more efficiently.

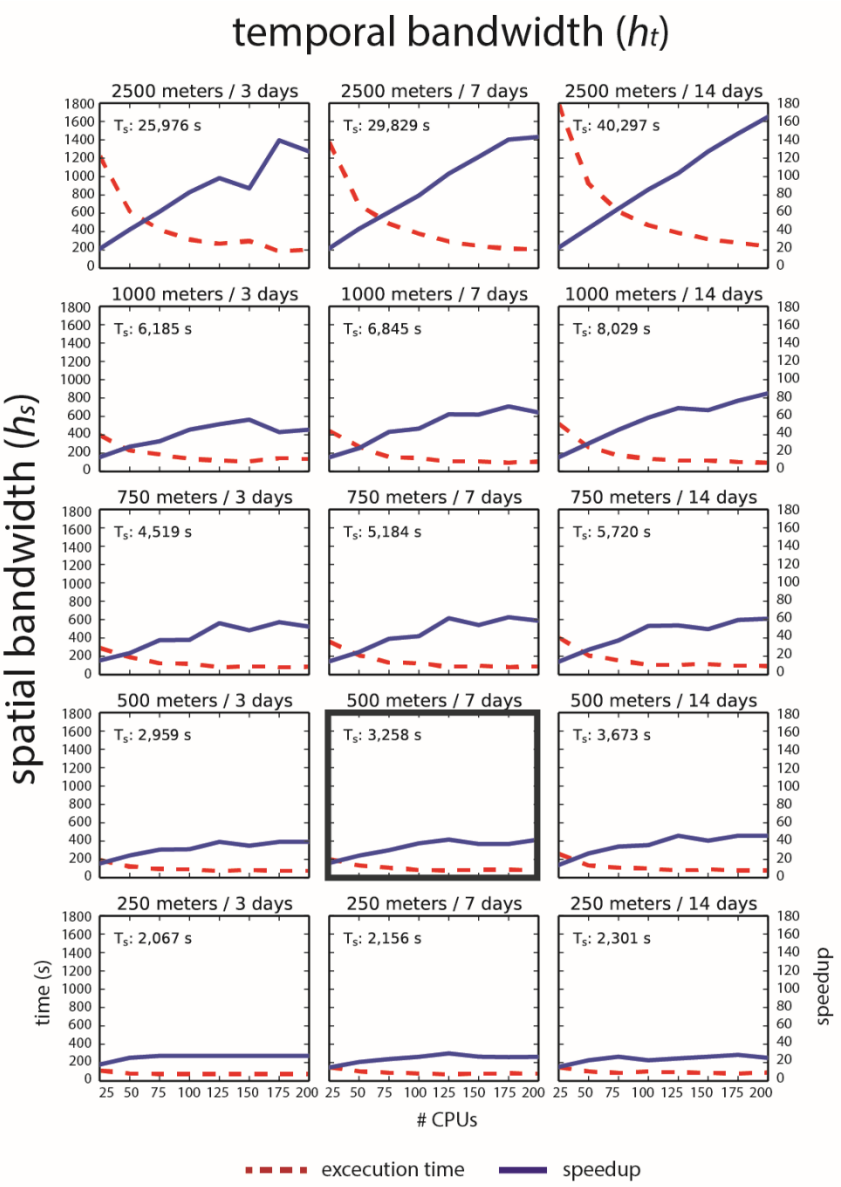


Figure 6: Acceleration performance (execution time and speedup) of space-time kernel density estimation algorithm over different spatial- and temporal bandwidths in response to alternative numbers of CPUs. $T_s$ is sequential time (using one CPU).

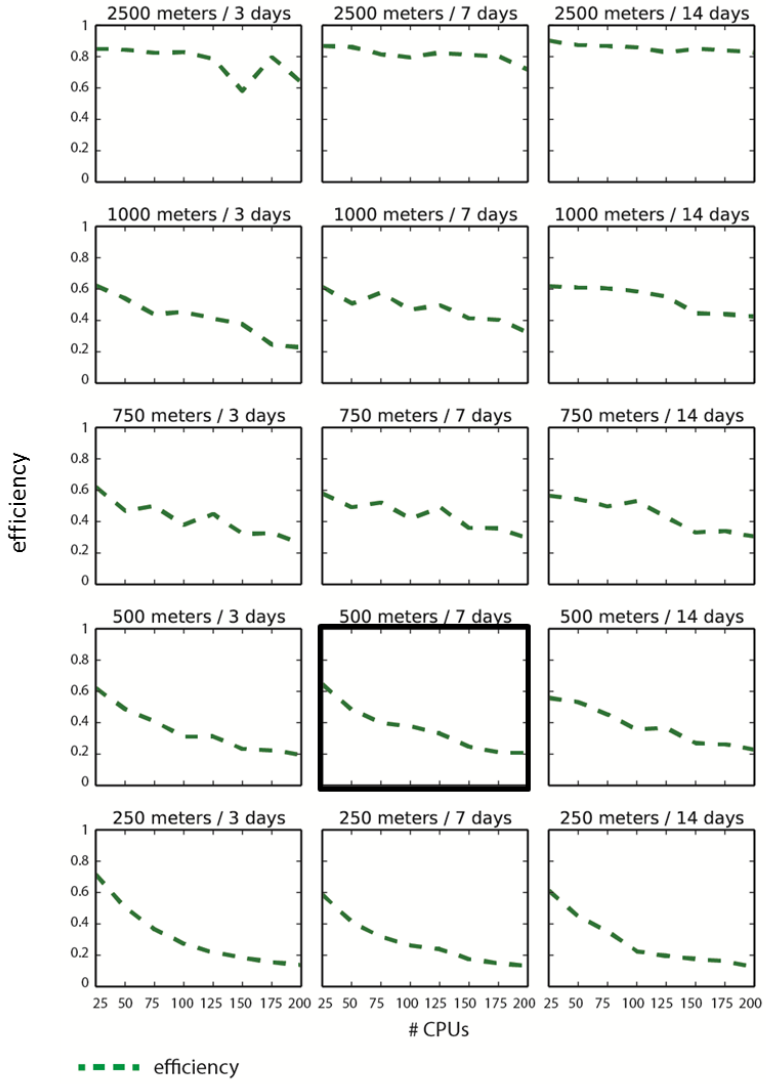



Figure 7: Efficiency of parallel space-time kernel density estimation algorithm over different spatial- and temporal bandwidths in response to alternative numbers of CPUs.

# *4.* Conclusions

Infectious diseases can spread very quickly under suitable conditions and timely monitoring is thus critical to reduce the risk of potential outbreaks. Space-time statistics facilitate the discovery of disease dynamics such as rate of spread and seasonal cyclic patterns, but these methods are computationally demanding, especially for large individual level data and at very fine resolutions. In this article, we develop an adaptive, space-time domain decomposition for parallel computation of the space-time kernel density, a computationally demanding space-time statistic. Our decomposition approach accounts for spatiotemporal heterogeneity in the data. We

illustrate the benefits of our method on a set of reported dengue cases in an urban environment of Colombia for the years 2010 and 2011 (2010 was an outbreak).

The parallel implementation of the STKDE reaches significant speedup compared to sequential counterparts. While sequential run may need up to 11 hours for STKDE algorithm, our parallel computing approach allows us to obtain STKDE results within minutes. This gives us more flexibility and capacity to efficiently and effectively detect the spatiotemporal clustering patterns in infectious disease data, represented by dengue fever in this study. Further, space-time kernel density estimates are visualized in an interactive 3D environment, which helps reveal space-time dengue fever clusters of different shapes and sizes.

Our approach dramatically reduces the computational effort required for the analysis of space-time patterns in epidemiological dataset at the individual level. This suggests that, -given a limited time budget-, space-time analysis can be conducted (1) at a much finer resolution, (2) for larger datasets, and for a (3) greater number of Monte-Carlo simulations, which would increase the statistical power of space-time statistics. Our methodology is portable to other individual space-time tests (e.g. Knox, Mantel, SaTScan) and areal level (LISA, SaTScan). These methodologies (and STKDE) have in common that they rely heavily on costly nearest-neighbor search and distance calculations. Our framework decreases this cost by reducing the search space and distributes the computation more evenly across several CPUs.

Our future work includes, but is not limited to: 1) applying our parallel approach to other space-time disease events; 2) testing other spatiotemporal domain decomposition strategies, including the use of k/d-trees or R-trees to reduce algorithm complexity, and to facilitate the parallel space-time analysis of disease datasets; 3) extending our approach to areal tests of space-time clustering, such as the temporal extension of the local Moran's I statistic and 4) improving our statistical methodology by using the spatiotemporal inhomogeneous K-function.

**References**


Andrzejewski, W., A. Gramacki, and J. Gramacki. 2013. Graphics processing units in acceleration of bandwidth selection for kernel density estimation. *International Journal of Applied Mathematics and Computer Science* 23 (4):869-885.

Anselin, L. 1995. Local indicators of spatial association-LISA. *Geographical analysis* 27 (2):93-115.

Armstrong, M. P., and R. Marciano. 1995. Massively parallel processing of spatial statistics. *International Journal of Geographical Information Systems* 9 (2):169-189.

Assuncao, R., M. Costa, A. Tavares, and S. Ferreira. 2006. Fast detection of arbitrarily shaped disease clusters. *Statistics in medicine* 25 (5):723-742.

Bader, M. 2012. *Space-filling curves: an introduction with applications in scientific computing*: Springer Science & Business Media.

Bailey, T., and Q. Gatrell. 1995. *Interactive Spatial Data Analysis*. Edinburgh Gate, England: Pearson Education Limited.

Bolibok, L. 2008. Limitations of Ripley K function use in the analysis of spatial patterns of tree stands with heterogeneous structure. *Acta Scientiarum Polonorum Silvarum Colendarum Ratio et Industria Lignaria* 7 (1):5-18.

Cali, S. 2010. *Historia del dengue en Cali. Endemia o una continua epidemia*: Cali: Secretaria de Salud Publica Municipal de Cali.

Carlos, H., X. Shi, J. Sargent, S. Tanski, and E. Berke. 2010. Density estimation and adaptive bandwidths: a primer for public health practitioners. *International Journal of Health Geographics* 9 (1):39.

Costa, M. A., R. M. Assunção, and M. Kulldorff. 2012. Constrained spanning tree algorithms for irregularly-shaped spatial clustering. *Computational Statistics & Data Analysis* 56 (6):1771-1783.

de Cali, A. d. S. 2008. Cali en cifras 2008. *Cali: Departamento Administrativo de Planeación*.

Delmelle, E., I. Casas, J. H. Rojas, and A. Varela. 2013. Spatio-temporal patterns of Dengue Fever in Cali, Colombia. *International Journal of Applied Geospatial Research (IJAGR)* 4 (4):58-75.

Delmelle, E., C. Dony, I. Casas, M. Jia, and W. Tang. 2014. Visualizing the impact of space-time uncertainties on dengue fever patterns. *International Journal of Geographical Information Science* 28 (5):1107-1127.

Delmelle, E., M. Jia, C. Dony, I. Casas, and W. Tang. 2015. Space-time visualization of dengue fever outbreaks. *Spatial analysis in health geography. Ashgate*.

Delmelle, E. M., H. Zhu, W. Tang, and I. Casas. 2014. A web-based geospatial toolkit for the monitoring of dengue fever. *Applied Geography* 52:144-152.

Demšar, U., and K. Virrantaus. 2010. Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science* 24 (10):1527-1542.

Ding, Y., and P. J. Densham. 1996. Spatial strategies for parallel spatial modelling. *International Journal of Geographical Information Systems* 10 (6):669-698.

Dixon, P. M. 2002. Ripley's K function. *Encyclopedia of environmetrics*.

Duczmal, L., A. R. Duarte, and R. Tavares. 2009. Extensions of the Scan Statistic for the Detection and Inference of SpatialClusters. In *Scan Statistics*, 153-177: Springer.

Eisen, L., and R. Eisen. 2011. Using geographic information systems and decision support systems for the prediction, prevention, and control of vector-borne diseases. *Annual Review of Entomology* 56 (1):41-61.
Epanechnikov, V. A. 1969. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications* 14 (1):153-158.
Gabriel, E. 2014. Estimating Second-Order Characteristics of Inhomogeneous Spatio-Temporal Point Processes. *Methodology and Computing in Applied Probability* 16 (2):411-431.
Gabriel, E., and P. J. Diggle. 2009. Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica* 63 (1):43-51.
Gabriel, E., B. Rowlingson, and P. Diggle. 2013. stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. *Journal of Statistical Software* 53 (2):1-29.
Graham, R. L. 1994. *Concrete mathematics:[a foundation for computer science; dedicated to Leonhard Euler (1707-1783)]*: Pearson Education India.
Grubesic, T. H., R. Wei, and A. T. Murray. 2014. Spatial Clustering Overview and Comparison: Accuracy, Sensitivity, and Computational Expense. *Annals of the Association of American Geographers* 104 (6):1134-1156.
Hohl, A., E. Delmelle, and W. Tang. 2015. Spatiotemporal domain decomposition for massive parallel computation of space-time kernel density. *ISPRS Annals of the Photographer, Remote Sensing and Spatial Information Sciences*.
Jacquez, G., D. Greiling, and A. Kaufmann. 2005. Design and implementation of a space-time intelligence system for disease surveillance. *Journal of Geographical Systems* 7 (1):7-23.
Kulldorff, M. 1997. A spatial scan statistic. *Communications in Statistics-Theory and methods* 26 (6):1481-1496.
Kulldorff, M. 2010. SaTScan-Software for the spatial, temporal, and space-time scan statistics. *Boston: Harvard Medical School and Harvard Pilgrim Health Care*.
Kulldorff, M., R. Heffernan, J. Hartman, R. Assunçao, and F. Mostashari. 2005. A space-time permutation scan statistic for disease outbreak detection. *PLoS medicine* 2 (3):216.
Kulldorff, M., and U. Hjalmars. 1999. The Knox method and other tests for space-time interaction. *Biometrics*:544-552.
Kwan, M.-P., I. Casas, and B. Schmitz. 2004. Protection of Geoprivacy and Accuracy of Spatial Information: How Effective Are Geographical Masks? *Cartographica: The International Journal for Geographic Information and Geovisualization* 39 (2):15-28.
Liu, Y., K. Wu, S. Wang, Y. Zhao, and Q. Huang. 2010. A MapReduce approach to G i*(d) spatial statistic. Paper read at Proceedings of the ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems.
Mantel, N. 1967. The detection of disease clustering and a generalized regression approach. *Cancer research* 27 (2 Part 1):209-220.
Mclafferty, S. 2015. Disease cluster detection methods: recent developments and public health implications. *Annals of GIS* (ahead-of-print):1-7.
Moran, P. A. 1950. Notes on continuous stochastic phenomena. *Biometrika*:17-23.
Nakaya, T., and K. Yano. 2010. Visualising crime clusters in a space-time cube: an exploratory data-analysis approach using space-time kernel density estimation and scan statistics. *Transactions in GIS* 14 (3):223-239.
Pfister, G. F. 2001. An introduction to the infiniband architecture. *High Performance Mass Storage and Parallel I/O* 42:617-632.
Restrepo, L. D. E. 2011. El plan piloto de cali 1950. *Bitácora Urbano Territorial* 1 (10):222-233.

Ripley, B. D. 1976. The second-order analysis of stationary point processes. *Journal of applied probability*:255-266.
———. 1977. Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B (Methodological)*:172-212.
Robertson, C., T. A. Nelson, Y. C. MacNab, and A. B. Lawson. 2010. Review of methods for space–time disease surveillance. *Spatial and Spatio-temporal Epidemiology* 1 (2):105-116.
Rogerson, P., and I. Yamada. 2008. *Statistical Detection and Surveillance of Geographic Clusters*. Boca Raton, Florida: CRC Press.
Takahashi, K., M. Kulldorff, T. Tango, and K. Yih. 2008. A flexibly shaped space-time scan statistic for disease outbreak detection and monitoring. *International Journal of Health Geographics* 7 (1):14.
Tang, W., W. Feng, and M. Jia. 2014. Massively parallel spatial point pattern analysis: Ripley's K function accelerated using graphics processing units. *International Journal of Geographical Information Science* (ahead-of-print):1-28.
Tango, T., and K. Takahashi. 2005. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics* 4 (1):11.
Turton, I. 2003. Parallel processing in geography. Paper read at Geocomputation.
Wang, S., and M. P. Armstrong. 2003. A quadtree approach to domain decomposition for spatial interpolation in grid computing environments. *Parallel Computing* 29 (10):1481-1504.
———. 2009. A theoretical approach to the use of cyberinfrastructure in geographical analysis. *International Journal of Geographical Information Science* 23 (2):169-193.
Wang, S., M. K. Cowles, and M. P. Armstrong. 2008. Grid computing of spatial statistics: using the TeraGrid for G i*(d) analysis. *Concurrency and Computation: Practice and Experience* 20 (14):1697-1720.
Widener, M., N. Crago, and J. Aldstadt. 2012. Developing a parallel computational implementation of AMOEBA. *International Journal of Geographical Information Science* 26 (9):1707-1723.
Wilkinson, B., and M. Allen. 2004. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (Second Edition)*. Upper Saddle River, NJ USA: Pearson Prentice Hall.
Yamada, I., and P. A. Rogerson. 2003. An Empirical Comparison of Edge Effect Correction Methods Applied to K-function Analysis. *Geographical analysis* 35 (2):97-109.