A Modification of Ripley’s K Function to Measure Aggregation About a Mass

Michael O. Smith\textsuperscript{1}, Jackson Ball\textsuperscript{2,3}, Benjamin B. Holloway\textsuperscript{2,3}, Ferenc Erdelyi\textsuperscript{4}, Gabor Szabo\textsuperscript{4}, Emily Stone\textsuperscript{1,2}, Jonathan Graham\textsuperscript{1} and J. Josh Lawrence\textsuperscript{2,3}

\textsuperscript{1}Department of Mathematical and Statistical Sciences, University of Montana, Missoula, MT 59812
\textsuperscript{2}Center for Structural and Functional Neuroscience, University of Montana, Missoula, MT 59812
\textsuperscript{3}Department of Biomedical and Pharmaceutical Sciences, University of Montana, Missoula, MT 59812
\textsuperscript{4}Laboratory of Molecular Biology and Genetics, Institute of Experimental Medicine, Hungarian Academy of Sciences, H-1450 Budapest, Hungary
Abstract

We present a methodology that detects spatial patterns of 3-dimensional point processes that include a mass within the study region. Spatial patterns such as clustering, randomness, or repulsion are considered in respect to the mass surface. Our method closely resembles Ripley’s K Function but is modified to discern the pattern about the mass surface. We walk though the definition and derivation of Ripley’s K Function and then follow this process to define the Modified K function. We develop this novel function according to the definition: the Modified K function times the intensity is the expected number of events within a distance h of a mass. Special consideration of edge effects is taken in order to make the function invariant to the location of the mass within the study region. Significance of spatial patterns is determined using Monte Carlo confidence envelopes similar to Ripley’s K Function. Simulations are performed to inform researchers how the Modified K function performs under different types of aggregation. Finally, we apply the Modified K function to neuroscience as a novel analysis tool by examining the spatial pattern of neurotransmitter release sites as events about a neuron. Supplemental materials for this article are available online.

Key Words: 3-Dimensions; Clustering; Point Process; Spatial Patterns; Spatial Statistics

1 Introduction

Ripley’s K Function is a tool used to characterize the strength of spatial dependence at multiple scales in a spatial point pattern. The K function has been widely used to identify clustering, randomness, or regularity among events in a spatial point pattern (Ripley 1977). Recent applications of Ripley’s K function have been to the field of biology, where the spatial organization of molecules can be described with a spatial point pattern (Lagache et al. 2013; Kellner et al. 2007; Jenei et al. 2009). Ripley’s K Function has been used in the field of neuroscience primarily to describe the spatial pattern of neurons in 3-dimensions (Jafari-Mamaghani et al. 2010; Hansson et al. 2013; Eglen and Wong 2008; Millet et al. 2011). This paper explores a modification to Ripley’s K Function to characterize the level of aggregation of a point pattern relative to a mass. Analysis of spatial aggregation relative to a mass is a novel application.

This work was motivated by the increasing recognition that communication between neurons depends on the fine architecture of presynaptic (neurotransmitter-releasing) and postsynaptic (neurotransmitter-sensing) elements in the brain. The spatial pattern of release sites about a neuron is of physiological significance to synaptic transmission but has not yet been described mathematically. Towards this end, image stacks of neurotransmitter-releasing objects relative to a neuron were rendered in 3-dimensions and analyzed using a modified Ripley’s K function. If we consider the objects as events,
these images can be reduced to a 3-dimensional spatial point pattern distributed around a neuron. The objective in this application is to ascertain whether the objects are aggregated about the surface of the neuron, and if so to what degree. Applying the usual Ripley’s K Function to this point pattern, ignoring the neuron body, would measure the spatial aggregation among the objects, but would not tell us whether these objects tend to be aggregated about neuronal surfaces. A modification to the K function is needed to assess whether the objects are aggregated about a neuron.

This paper consists of five additional sections. Since the motivation for this new method is derived from Ripley’s K-function, section 2 reviews the construction and method of Ripley’s K, as well as the definition of a spatial point process. An example of the use and interpretation of Ripley’s K Function on the Longleaf pine data, given in Cressie (1993), is presented. Section 3 introduces the Modified Ripley’s K function for measuring spatial aggregation about a mass in a point pattern. Construction of this method closely mimics the construction of Ripley’s K, but differences are noted and discussed. Section 4 shows results of a simulation study assessing how well this new method characterizes the spatial aggregation under various known conditions. Then, by applying the Modified Ripley’s K function to neuroscience, analysis of events about a neuron is conducted in section 5. Finally, a discussion of important topics and future studies are discussed.

2 Ripley’s K Function

2.1 Spatial Point Pattern

Before we discuss Ripley’s K Function, we first consider the definition of a spatial point pattern. Spatial point patterns are a type of spatial data, consisting of an arrangement of point locations \((s_1, s_2, ..., s_n)\) on some random set \(R \subset \mathbb{R}^n\) where \(n\) is typically 2 or 3 (Schabenberger and Gotway 2005). The set \(R\) is taken to be some study region of interest. Often the study region is rectangular but can be any designated closed region. The study region is also not limited to 2-dimensions. As we will see in the application presented in this paper, the study region can have a third dimension.

The point locations, which we will refer to as events, denoted as \(s_i\), for \(i = 1, ..., n\) are vectors indicating the location of the event in \(R\). In \(\mathbb{R}^2\), \(s_i = (s_{i1}, s_{i2})^T\), where the components of \(s_i\) are simply the x and y coordinates, respectively. Of course if the dimension of \(R\) is greater than two, then \(s_i\) would have more than two components or coordinates. Examples of events could be locations of longleaf pine trees in a forest, locations of lung cancer patients within a country, or crime locations in a city. We use the word “event” for \(s_i\) to distinguish these locations of interest from arbitrary point locations in \(R\) (Cressie 1993). Hence, we have “events” which are observed occurrences of interest and “points” which are arbitrary locations in \(R\).
A typical objective with a point pattern is to characterize its pattern of events. In general, there are three patterns of interest: clustering, complete spatial randomness (CSR), or regularity. Figure 1 illustrates these three spatial point patterns. From left to right we have a clustered, CSR, and regular point pattern. There is a natural ordering of these patterns where CSR is at the midpoint of the continuum and on either side of it the point process is clustered or regular.

Ripley’s K Function quantifies departures from CSR at multiple spatial scales. In order to use Ripley’s K-function, we make some assumptions. The mean and dependence structure of point patterns are characterized by their first and second-order properties. First-order effects arise when there are variations in the average number of events throughout the study region. In other words, we would observe a larger number of events in one subregion of the study region compared to another disjoint subregion. When this occurs, we say the point pattern is inhomogenous. Second-order effects arise when locations of (or distance between) pairs of events are dependent on each other. In other words, we would observe a tendency for event locations to “attract” or “repulse” each other (Bailey and Gatrell 1995).

These concepts are illustrated in the following example given by Bailey and Gatrell (1995, p. 32). Suppose you have iron shavings as events and you scatter them on a sheet of paper at random. If this is done in a completely random fashion, the resulting point pattern would exhibit no first or second-order effects. There is no variation in the average number of events throughout the study region (first-order effects), and all event locations are independent from other event locations (second-order effects). Now suppose you place weak magnets under the sheet of paper and again drop the iron shavings on the paper at random. Since the shavings will be attracted toward the location of the magnets under the paper the point pattern would exhibit a first-order effect and no second-order effect. The first-order effect occurs since we observe deviations from a constant average
number of events as we move throughout the study region. Now remove the magnets and instead, weakly magnetize the iron shavings and again drop the shavings on the paper at random. Now the magnetized iron shaving locations tend to “follow” other shaving locations resulting in a point pattern with second-order effects and no first-order effects. The second-order effects are due to the dependency between event locations. Lastly we could place the weak magnets under the paper and weakly magnetize the iron shavings in order to observe a point pattern arising from a mixture of first and second-order effects.

The distinction between first and second-order effects is important since, later in this paper we assume point patterns to arise only from second-order effects. If we assume no first-order effects arise in a spatial point pattern the pattern is said to be stationary. Formally, a point pattern \((\mathcal{R})\) is stationary if all statistical properties of a subregion of \(\mathcal{R}\) are invariant to arbitrary translation about \(\mathcal{R}\). Furthermore, a point pattern is isotropic if the same invariance occurs under rotation (Diggle 1983). By definition, isotropy implies stationarity. Also, if a point pattern is stationary and isotropic it is said to be homogeneous.

First-order effects can be modeled by the intensity function. The first-order intensity measures the mean number of events per unit area (or volume in 3D). Mathematically the intensity of a point pattern is defined as

\[
\lambda(s) = \lim_{ds \to 0} \left\{ \frac{E[Y(ds)]}{ds} \right\}
\]

where \(ds\) is a small region about the point \(s\), \(ds\) is the area of this region, and \(Y(\cdot)\) is a random variable representing the number of events in some region. For a homogeneous (stationary and isotropic) point process we would consider the intensity to remain constant throughout \(\mathcal{R}\). In other words, for a homogeneous point pattern the first-order intensity does not depend on spatial location \((\lambda(s) = \lambda)\) and we observe no first-order effects.

Second-order effects can be modeled by a second-order intensity function, but is difficult to interpret with much depth (Schabenberger and Gotway 2005). For stationary point patterns, the second-order intensity function only depends on direction and distance between pairs of events. If we assume an isotropic point pattern the second-order intensity function depends purely on distance between pairs of points. Again we note that for a stationary or isotropic point pattern we assume no first-order effects arise. In replacement of the second-order intensity function, a more intuitive model for second-order effects is Ripley’s K Function. Ripley’s K is an effective tool for measuring or quantifying spatial dependence over multiple spatial scales as opposed to nearest neighbor methods which consider only small scales.

For a homogenous point pattern, Ripley’s K Function (Bailey and Gatrell 1995) is defined by:
\[ \lambda K(h) = E(\text{number of events within distance } h \text{ of a single arbitrary event}) \]  

(2)

where \( \lambda \) = the intensity or the mean number of events per unit area, assumed constant throughout \( \mathcal{R} \), and \( E \) is the expectation function. Note that when we say "single arbitrary event" we are only considering one event within the point pattern. To make this clear, Figure 2 depicts the right hand side of the equation. A single arbitrary event is chosen, we count all other events within a distance \( h \), and average these counts over all such events in the pattern as an estimate of this expectation.

Let the area of \( \mathcal{R} \) be \( R \); then the expected number of events in \( \mathcal{R} \) is \( \lambda R \). To generalize this to all events, note that if we multiply the expectation in (2) by the average number of events in the study region, \( \lambda R \), this gives the expected number of event pairs within distance \( h \) of one another. Multiplying both sides of (2) by \( \lambda R \) yields:

\[ \lambda R \cdot \lambda K(h) = \lambda R \cdot E(\text{number of events within distance } h \text{ of a single arbitrary event}) \]  

(3)

which implies

\[ \lambda^2 R K(h) = E(\text{number of event pairs at most } h \text{ apart}). \]  

(4)

Graphically, if we imagine circles of radius \( h \) around each event, we search for other events within that circle. Technically we would count each "link" between each event pair.
within $h$ twice. In Figure 3, an estimate of $E$ (number of event pairs at most $h$ apart) would be four since the two “links” are each counted twice. We will account for the “double counting” when we estimate the expectation.

Figure 3: Event pairs illustrating how Ripley’s K Function expectation is estimated. All figure circles have the same radius $h$, thus at this $h$, the estimate counts two event pairs.

Now let $d_{ij}$ be the distance between the $i$th and $j$th observed events in $\mathcal{R}$. Then we define the indicator function

$$I(d_{ij} \leq h) = \begin{cases} 1 & \text{if } d_{ij} \leq h, \\ 0 & \text{otherwise}. \end{cases}$$

(5)

An estimate of the expected number of event pairs at most $h$ apart is $\sum \sum_{i \neq j} I(d_{ij} \leq h)$. In other words, the double summation term is the observed number of such event pairs within distance $h$ of one another. Notice that when we let the indexes of the summations be $i \neq j$ we are again counting the event pairs (or “links”) twice. Thus, a suitable estimate of $K(h)$ is given by

$$\hat{K}(h) = \frac{1}{\lambda^2 R} \sum \sum I_h(d_{ij} \leq h).$$

(6)

Next we need to consider the effects of the study region boundary on the estimator; such effects are called edge effects. The current estimator of $K(h)$ does not take into account event pairs where one of the events is outside $\mathcal{R}$. Thus some type of correction
or extrapolation must be implemented to correct for this edge effect. Consider a circle centered on event $i$ that contains $j$. Then let $w(d_{ij})$ be the proportion of the circumference of this circle which lies within $\mathcal{R}$ (Figure 4). $w(d_{ij})$ can be thought of as the conditional probability that an event is observed in $\mathcal{R}$, given that it is at most a distance $d_{ij}$ from a given event $i$. Notice that $w(d_{ij})$ is not a function of $h$ but a function of $d_{ij}$. Also, notice that this edge correction is only valid if the point process is isotropic. To correct for the edge effect we incorporate $w(d_{ij})$ into (6) as follows:

$$\hat{K}(h) = \frac{1}{\lambda^2 R} \sum_{i \neq j} \sum I(d_{ij} \leq h) \frac{w(d_{ij})}{w(d_{ij})}. \quad (7)$$

Note that for an event not near the edge, $w(d_{ij}) = 1$ and no edge correction is implemented. However, for an event near the edge, $w(d_{ij})$ may be less than 1, inflating its contribution to $\hat{K}(h)$. Lastly we must estimate $\lambda$, the mean number of events per unit area. Since we assume the point pattern is homogeneous and the intensity is constant, then a simple estimate is $\hat{\lambda} = n/R$, where $n$ is the number of events in $\mathcal{R}$, and $n/R$ is the number of events per unit area in $\mathcal{R}$. Thus our final estimate of $K(h)$ is:

$$\hat{K}(h) = \frac{R}{n^2} \sum_{i \neq j} \sum I(d_{ij}) \frac{w(d_{ij})}{w(d_{ij})}. \quad (8)$$
Figure 5: A graphical notion of Ripley’s K Function. Each event is visited and concentric circles of many radii are created. Events within these circles are then counted as an event pair. The estimate of Ripley’s K counts these event pairs.

This estimate can be difficult to understand, so we will explain the notion of the K function graphically. Consider every point in the point pattern visited. When a single point is visited, consider concentric circles of radius $h$ placed around this visited event. If other events fall within these circles the event pair is counted for that value of $h$. When all of these event pairs have been counted for a single value of $h$, the number is then scaled by $R/n^2$. As an example, Figure 5 has two events visited. The visited event on the left has one event within five units (and four units.) The visited event on the right has four events that are within five units. Note that this explanation ignores the edge effect.

2.2 Complete Spatial Randomness and $\hat{L}(h)$

$\hat{K}(h)$ can be plotted against values of distance $h$ but little inference about the level of spatial dependence can be made from such a figure. $\hat{K}(h)$ must be compared to $K(h)$ under an assumption of randomness or when there is no spatial dependence. Randomness of a spatial point pattern is called complete spatial randomness, or CSR. To explain CSR in greater detail, we follow the notation and ideas seen in Bailey and Gatrell (1995). As previously stated, $\mathcal{R}$ represents the entire study region. Let $\mathcal{A}$ be a subregion of $\mathcal{R}$, and the random variable $Y(\mathcal{A})$ be the number of events within the subregion $\mathcal{A}$. Then a point pattern can be described by the collection of random variables $\{Y(\mathcal{A}), \mathcal{A} \subseteq \mathcal{R}\}$. CSR implies that random variables $Y(\mathcal{A}_i)$ and $Y(\mathcal{A}_j)$ are independent for any choice of $\mathcal{A}_i$.
and $A_j$. Furthermore, CSR implies that the probability distribution of $Y(A)$ is Poisson with mean $\lambda A$, where $A$ is the area of $A$. An equivalent notion of CSR is that events are independently and uniformly distributed over $R$. This implies that each event’s location is independent of that of other events and each area is equally likely to possess an event.

One nice property of Ripley’s K Function is the existence of a theoretical value under CSR. Under CSR the expected number of events within a radius $h$ of a single arbitrary event is the intensity times the area of the circle with radius $h$ or $\lambda \pi h^2$. Applying this to equation (2), and ignoring edge effects we would thus expect $K(h) = \pi h^2$ under CSR. Again, we emphasize that a point pattern under CSR exhibits no spatial dependence. If $K(h) > \pi h^2$, then more events arise within a distance $h$ from a single event than expected under CSR; this indicates clustering. On the other hand if $K(h) < \pi h^2$, this indicates regularity. Thus we would like to compare $\hat{K}(h)$ to $\pi h^2$ to assess the type of spatial dependence present. A common way to perform this comparison is through the function

$$L(h) = \sqrt{\frac{K(h)}{\pi}} - h. \quad (9)$$

Positive values of $L$ indicate clustering since we are observing more event pairs within $h$ compared to a point pattern under CSR. Conversely, negative values of $L$ indicate regularity within the point pattern. Letting $\hat{L}(h) = \sqrt{\frac{\hat{K}(h)}{\pi}} - h$, we expect $\hat{L}(h) = 0$ under CSR. Note that an equivalent comparison of $\hat{K}$ to $K$ under CSR would be to look at $\hat{K}(h) - \pi h^2$. Plots of both $\hat{K}$ and $\hat{L}$ will be presented later using the longleaf pine data.

We next assess how positive (or how negative) $\hat{L}(h)$ must be in order to conclude significant clustering (or regularity.) This is traditionally done via Monte Carlo simulations (simulation envelopes) where point patterns of the same size $n$ as the number of observed events in $R$ are generated under CSR and $\hat{L}(h)$ is computed for each simulation (Schabenberger and Gotway 2005, pp. 87-88). Conceptually, the generation of point patterns under CSR amounts to “throwing down events” on the study region completely at random and then calculating $\hat{L}$ for each simulated process. If we let $\hat{L}^s(h)$ be the $s^{th}$ simulated $\hat{L}$ function under CSR, $s = 1, ..., m$ we can then find the $\alpha/2$ and $1 - \alpha/2$ percentiles of $\hat{L}^s(h)$ for each $h$ according to some $\alpha$ level. These percentile functions will be referred to as the lower and upper envelopes. Notationally the lower envelope at distance $h$ is

$$\mathcal{L}(h) = \hat{L}^s_{\alpha/2}(h) \quad (10)$$

which is the $\alpha/2$ percentile of all the simulated $L$ functions. Similarly the upper envelope at distance $h$ is
\[ U(h) = \hat{L}_{1-\alpha/2}(h) \]

which is the \( 1 - \alpha/2 \) percentile of all the simulated \( L \) functions. A plot of the three
functions \( \hat{L}(h), \mathcal{L}(h), \) and \( U(h) \), will now give us a graphical way to determine the scales
at which our observed point pattern exhibits clustering, randomness, or regularity.

### 2.3 Longleaf Pine Data

To illustrate how Ripley’s K Function might be used in practice, consider the longleaf pine
data presented in Cressie (1993). These data consist of 584 longleaf pine tree locations
and diameter at breast height in 4 hectares of land in the Wade Tract in Georgia, USA.
The study region was chosen since there is little elevation change and trees were only
sampled if their height was greater than 2cm at breast height. This information helps
us assume stationarity, since the study region most likely does not have any large varia-
tion in intensity. The gentle topography of the region also helps to justify the stronger
assumption of isotropy, since the covariance structure of the events is most likely due to
distance between pairs and not the direction between pairs. The point pattern is shown
in Figure 6, where each tree is marked by a cross. We can see that intensity is roughly
constant across the study region.

Figure 6 also includes a plot of \( \hat{L}(h) \) which exhibits all positive values. Significances
of the \( \hat{L}(h) \) values are determined by the confidence envelopes also shown. To create
envelopes we simulate point patterns under CSR using the same study region and with
583 events since this is how many longleaf pine trees were observed in the original point
pattern. This was done 1000 times to give envelopes at the \( \alpha = .05 \) level. The significance
of clustering is confirmed by these envelopes since the estimated L-function is well above
the two envelopes.
Figure 6: Longleaf pine point pattern illustrating locations of trees among a study region. Also, the estimated L function with Monte Carlo confidence envelopes. The estimated L falls above of the confidence envelopes at nearly all scales, indicating the longleaf pine tree point pattern to be significantly clustered at all scales.

3 The Modified K Function

3.1 Definition and Estimation

The longleaf pine data consist only of event locations whose spatial pattern can be analyzed with Ripley's K Function. Now consider a large object or mass in the study region along with events distributed around the mass. An example might be a nuclear power facility within a neighborhood with events defined as locations of deceased birds. In this example, one might hypothesize that event locations are spatially dependent on the location of the mass. If the mass was small relative the the study region, a single point could represent the mass. But, if the mass is large relative to the study region and there is an interest in clustering at small scales about the mass, a single point representation of the mass might be misleading. If the mass is not uniform in shape, the distance from the single point mass representation to the mass’s boundary would vary. Thus, distances found would not represent events about the mass’s boundary appropriately. Instead of considering a single point mass representation, we consider the mass’s boundary which could be represented by a closed surface or many points estimating the closed surface, typically in 2 or 3-dimensions. In our case, we want the mass to be “solid" and not part of the study region. This allows events to only fall outside the mass within some study region.
Figure 7: Example of a point pattern with a mass included. The band is created by “searching” at distance $h$ away from the mass surface. Only one band is shown in figure, but in practice many are created as $h$ varies. The Modified K function counts events within each band, and then compares that count with the expected number of events based on the area of each band.

When it is appropriate to consider a mass within the study region, and spatial aggregation about that mass is of concern, methods similar to Ripley’s K Function can be used. To make this modification we consider searching for events within a distance $h$ of the mass (Figure 7). When we do this we create a “band” about the mass that has width $h$. These bands vary as a function of $h$ and are similar to the circles placed around events in Ripley’s K. Figure 7 illustrates one such band about the mass. Suppose that the point pattern is isotropic. We define the Modified K function as:

$$\lambda K_M(h) = E(\# \text{ of events within distance } h \text{ of a mass})$$

(12)

where $\lambda$ is the intensity. This definition differs from Ripley’s K only in the expectation
since we are now considering all distances to be “mass to event” rather than “event to event”.

We estimate $K_M$ in a similar way as $K$. Let $d_i$ be the nearest distance between the $i$th event and the mass and let

$$I_h(d_i \leq h) = \begin{cases} 1 & \text{if } d_i \leq h, \\ 0 & \text{otherwise}. \end{cases}$$

(13)

If we sum (13) over all $i = 1, \ldots, n$ where $n$ is the number of events in the study region $\mathcal{R}$, then we are counting the number of events within each band of width $h$. This summation is an estimate of the expectation in (12). Note that as $h$ varies the bands are nested, which makes $\sum_{i=1}^n I_h(d_i \leq h)$ a cumulative count of events. Note that $d_i$ is defined as the nearest distance to the mass. A single event has many distances to the boundary of the mass; $d_i$ is the smallest of those distances. Since we assume the point pattern to be isotropic with constant intensity, we again estimate $\lambda$ by $n/R$. It should be noted that $R$ is the area of only the study region which does not include the mass. Thus, our estimate of $K_M$ is

$$\hat{K}_M(h) = \frac{R}{n} \sum_{i=1}^n I_h(d_i \leq h).$$

(14)

Now consider an edge correction for $K_M$. The correction is similar to that of Ripley’s $K$ where we consider the proportion of a circle’s circumference inside $\mathcal{R}$. With $K_M$ we have bands about the mass. Let $w(d_i)$ be the proportion of the band’s area inside $\mathcal{R}$. One might also consider using the proportion of the band’s outer circumference inside $\mathcal{R}$. Then for each $i$ we scale the value of $I(d_i \leq h)$ by it’s corresponding $w(d_i)$. The estimate of $K_M$ becomes

$$\hat{K}_M(h) = \frac{R}{n} \sum_{i=1}^n \frac{I_h(d_i \leq h)}{w(d_i)}.$$  

(15)

This edge correction effectively extrapolates the count of events outside the study region, making $\sum_{i=1}^n \frac{I_h(d_i \leq h)}{w(d_i)}$ invariant to the location of the mass about the study region. Moving the mass and its relative point pattern about the study region is essentially a translation of the mass and events. When this translation is performed different values of $n$ are found, since some events will fall outside the the study region depending on the translation. As a result, $n$ also needs to be adjusted for the edge. Let $n_e$ = the number of events within the largest distance $h$ from the mass, adjusted by the edge correction. Notationally,
\[ n_e = \sum_{i=1}^{n} \frac{I_h(d_i \leq \max h)}{w(d_i)}. \] (16)

This correction of \( n \) will allow the values of \( \hat{K}_M(h) \) to be roughly invariant to the location of the mass within \( \mathcal{R} \). We should note that \( n_e \) is solely an adjustment to the scale of our estimate. Later, when we simulate point patterns under CSR, we use \( n \) events and not \( n_e \) events. It should also be noted that the values of \( h \) are chosen by the researcher. With these edge adjustments, our final estimate of \( K_M \) is

\[ \hat{K}_M(h) = \frac{R}{n_e} \sum_{i=1}^{n} \frac{I_h(d_i \leq h)}{w(d_i)}. \] (17)

### 3.2 Modified K under CSR

Plots of \( K_M(h) \) are not very meaningful unless we compare the values to what we expect \( K_M \) to look like under CSR. This is a difficult task since the mass could take on shapes that are irregular. Shapes such as circles or rectangles would make this task easier. Consider the mass to be circular with radius \( r \) in a 2-dimensional study region. Then the “bands” about the mass will be donut shaped with a width of \( h \). The area of this band is computed as \( \pi(h + r)^2 - \pi r^2 \), so that using equation (12):

\[ \lambda K_M(h) = \lambda[\pi(h + r)^2 - \pi r^2]. \] (18)

Thus, under CSR, \( K_M(h) = \pi(h + r)^2 - \pi r^2 \) when the mass is a circle of radius \( r \). In this case, we can scale \( \hat{K}_M \) in order to create a modified L function as with Ripley’s \( \hat{K} \) function. Similarly if the mass is a rectangle, the area of the bands can be computed analytically and \( \hat{K}_M \) can be scaled. If the true area of bands can be found, one should utilize this information. However, if the mass is irregular and band areas are difficult or impossible to find, the value of \( K_M \) under CSR must be estimated.

To estimate \( K_M \) under CSR, we simulate point patterns under CSR. This is conceptually equivalent to randomly distributing events in the study region, but outside the mass. Each time a random point pattern is created \( \hat{K}_M(h) \) is calculated. Let \( \hat{K}_M^s(h), s = 1, 2, ..., m \) be the \( m \) simulated functions under CSR. We can then take the median (or mean) value of all the \( \hat{K}_M^s(h) \) for each \( h \) to be the estimated value of \( K_M(h) \) under CSR. Like Ripley’s K, the number of events we randomly distribute is dependent on \( n \), the number of events observed in \( \mathcal{R} \).

We can also use the \( \hat{K}_M^s(h) \) values, \( s = 1, 2, ..., m \), to create confidence envelopes according to some \( \alpha \) level. Let the \( \hat{K}_M \) lower envelope be the \( \alpha/2 \) percentile of the
\( \hat{K}_M^*(h) \) values at each \( h \). Similarly, let the \( K_M \) upper envelope be the \( 1 - \alpha/2 \) percentile of the \( \hat{K}_M^*(h) \) values. The \( \hat{K}_M(h) \) function and two \( \hat{K}_M \) envelopes can be scaled by the simulated estimate of \( K_M(h) \) under CSR to create \( L_M \) functions. Notationally,

\[
\hat{L}_M(h) = \hat{K}_M(h) - \text{median}[\hat{K}_M^*(h)] \tag{19}
\]
\[
L_M \text{ Upper Envelope} = K_M \text{ Upper Envelope} - \text{median}[\hat{K}_M^*(h)] \tag{20}
\]
\[
L_M \text{ Lower Envelope} = K_M \text{ Lower Envelope} - \text{median}[\hat{K}_M^*(h)]. \tag{21}
\]

Positive values of \( \hat{L}_M(h) \) indicate clustering about the mass since we are observing more points within the bands than would be expected under CSR. Negative values of \( L_M(h) \) indicate a repulsion from the mass since we are observing fewer events within the bands relative to CSR. The notion of repulsion was not used in the explanation of Ripley’s \( K \). With Ripley’s \( K \), negative \( L \) values indicated regularity within the point pattern. In our case, use of the term “regularity” among the mass is not clear. Instead we think of the events avoiding the mass if we conclude repulsion. By viewing the plots of \( \hat{L}_M(h) \) with envelopes, the significance of the negative or positive values of \( \hat{L}_M(h) \) can be determined.

4 Modified K Function Simulations

4.1 Simulations and Figures

To provide insight into how the Modified K function performs, a simulation study was performed. Simulated or contrived point processes were generated. Simulations were performed using a 2-dimensional point process (instead of 3-dimensions) to increase clarity of discussion. The processes were produced on a 1 unit by 1 unit study region. A mass is represented by a single point in the center of the study region (located at \((0.5, 0.5)\)). Using a single point instead of a mass increases clarity and does not seem to effect results. Three types of point processes were used. We will refer to the three processes as patterns 1, 2, and 3. Pattern 1 is the obvious first choice; a random point process. Computationally this implies sampling the \( x \) and \( y \) coordinates from two uniform distributions. Pattern 2 is a clustered point process. This process is produced by sampling distances from the mass using an exponential distribution with mean distance \( \lambda^{-1} \). If not otherwise stated, distances were sampled from an exponential distribution with mean \( 2^{-1} \) (\( \lambda = 2 \)). Direction from the mass was random. This creates a process where events are more likely to occur near the mass. As \( \lambda \) increases the point pattern exhibits a higher degree of clustering about the mass. Pattern 3 is also a clustered process but in a donut-like fashion. Events were sampled randomly but forced to occur between .1 and .2 units away from the mass, thus resulting in a “donut of events” around the mass. Figure 8 shows the three processes.
Figure 8: Point processes used in simulations. Study region is 1 unit by 1 unit with the mass marked as an “x”. Pattern 1 is a random point pattern, while Pattern 2 and 3 are clustered patterns. It should be noted that in the point processes seen here, the number of events \( n \) is 100. Some simulated point patterns used different values of \( n \), but were created with the same type of random or clustered fashion. These 3 point patterns are examples of point patterns used in the center graph \((n = 100)\) case of Figure 9.

The first simulation was performed with different numbers of events. \( n \) was set to be 10, 100, and 500. The resulting average \( \hat{L} \) functions and envelopes based on 1000 simulations are shown in Figure 9. In all cases of \( n \), the \( \hat{L} \) functions describe the patterns appropriately. Pattern 1, the random point process, remains within the envelopes for all values of \( h \). \( \hat{L} \) for Pattern 2, a clustered process, immediately exceeds the envelopes to indicate clustering about the mass. And pattern 3, a clustered process where all events occur between 0.1 and 0.2 units from the mass, increases to a spike between .1 and .2. These characteristics of the \( \hat{L} \) functions occur with all simulations hereafter. Considering the number of events, \( n \); as \( n \) increases the \( \hat{L} \) functions and envelopes become less noisy as expected. Also, as \( n \) increases the envelope values narrow toward 0.

Figure 10 was created using two maximum distances from the mass, 0.3 and 0.6. In other words, the K function only performed calculations up to the respective maximum distance. Again, the L functions describe patterns 2 and 3 appropriately. In all cases with the Modified L function, the function and envelopes terminate at 0 for the maximum value of \( h \) chosen. This is an artifact of the Modified L function scaling and is similar that of Ripley’s K Function. In both aggregated patterns in 10, the scale of aggregation is inflated when the max distance is increased, especially with pattern 3.
Figure 9: Modified L with different number of events used. As number of events increases, the noise among the $\hat{L}$ functions and envelopes are dampened.

Figure 10: Comparing the L function with different choices of the maximum distance. Increasing the maximum distance with Pattern 3 inflates the peak magnitude of the Modified $L$ values.
The second simulation is shown in Figure 11. 1000 simulations were used to show the effect of moving the mass from the center of the 1 unit by 1 unit study region, \((0.5, 0.5)\), toward a corner of the region at, \((0.3, 0.3)\). Doing this heightens the effect of the edge correction built into the Modified \(\hat{K}\) and \(\hat{L}\) functions. Patterns 1 and 3 exhibit no bias and are identical when the mass is moved. Pattern 2 exhibits some bias by showing more clustering when the center is at \((0.3, 0.3)\). Also the envelopes become slightly more narrow and approach zero more abruptly at \(h = 0.6\).

Figure 12 only uses point processes similar to that of pattern 2. Like in pattern 2 explained above we use the exponential distribution to generate distances from the mass for each event. Different parameter \(\lambda\) values are used to vary the degree of clustering about the mass. Specifically \(\lambda\) was set to 1, 5, and 10, where smaller choices of \(\lambda\) correspond to less aggregation about the mass. The simulation shows that the \(\hat{L}\) functions illustrate greater clustering when \(\lambda\) is increased as expected. \(\hat{L}\) functions and envelopes are means of 1000 simulations.

![Figure 11: Effect of invoking an edge effect while using the Modified L function. Little to no bias is seen when an edge effect is invoked, indicating the edge correction of the Modified K function is stable.](image)

Figure 11: Effect of invoking an edge effect while using the Modified L function. Little to no bias is seen when an edge effect is invoked, indicating the edge correction of the Modified K function is stable.
Figure 12: Effect of increased clustered patterns. As the point pattern becomes more clustered (more events near mass), the magnitude of the Modified $\hat{L}$ increases.

The results from the simulations illustrate that the Modified L function is stable and measures aggregation well when a mass is present in the study region. The correction to the Modified L function, $n_c$, almost completely removes any bias when an edge effect is present. This is very useful since the mass might be located near the edge. The modified L function works very similar to Ripley’s K and L functions, but measures a different kind of aggregation. Notice that when using pattern 3, the “aggregated donut”, there are no events located past a distance of $0.2$ from the mass. Yet, the function still indicates significant clustering beyond $0.2$. The function does however immediately begin to taper back toward zero.

It is possible to compare the quantity of aggregation by using methods similar to the index of association found in Fajardo et al. (2006). Ratios of the areas (or distances) between the envelopes and the L function can be found. However, as with figure 10, one should consider using the same maximum distance, $h$, before comparing. Comparing two different aggregated patterns like pattern 2 and pattern 3 could be misleading also. These two patterns are different kinds of clustering and are difficult to conclude which is more clustered about the mass.
5 Application of Ripley’s K Function to Cholinergic Neurotransmission

5.1 Rationale and implications for investigating the spatial distribution of cholinergic terminals in the hippocampus

The release of the excitatory neurotransmitter acetylcholine (ACh) into the hippocampus critically modulates learning and memory through the activation of postsynaptic receptors (Hasselmo and Giocomo 2006; Ruivo and Mellor 2013). Cholinergic dysfunction in the hippocampus has been implicated in diverse disease states, including Alzheimer’s disease, epilepsy, and autism spectrum disorders. The source of ACh arises from fibers of the medial septum-diagonal band of Broca (MS-DBB), which diffusely innervate all areas of the hippocampus. At a cellular and synaptic level, there is a growing debate as to whether some cells are specifically targeted for cholinergic neuromodulation. Therefore, understanding the spatial relationships between cholinergic fibers and their targets could yield important insight into disease states. Release of acetylcholine from synaptic terminals, or boutons, are thought to activate neurons through two different modes that depend on the spatial relationship to their neuronal targets. At one extreme, cholinergic transmission employs a “classical” mode in which a synapse is targeted to the immediate vicinity (within 50 nm) of membrane-bound receptors that are associated with the synapse. This mode, called point-to-point transmission, entails the release of ACh at a high concentration to activate synaptic receptors within 1 ms. At the opposite extreme, ACh is released at some distance, diffuses in 3-dimensions, and ultimately decays to a low concentration that activates high affinity receptors distributed along the surface of the neuron. This mode is called volumetric transmission (Vizi et al. 2004). These two modes are not mutually exclusive. However, the concentration of ACh that reaches a neuron depends upon the precise spatial arrangement and density of ACh release sites relative to the neuronal surface. Importantly, the enzyme acetylcholinesterase (AChE) rapidly degrades ACh in the extracellular space, which underscores the complex temporal and spatial relationships of ACh release sites relative to neurons. AChE inhibitors, the only current therapy for treatment of Alzheimer’s Disease, block the degradation of ACh, thereby making volumetric transmission more efficacious by altering spatiotemporal aspects of diffusion between ACh release sites and postsynaptic neurons.

The spatial organization of ACh release sites relative to neuronal types is poorly understood. Using an antibody to the vesicular acetylcholine transporter (vAChT), there is some evidence that ACh release sites are preferentially clustered at distinct subtypes of neurons (Dougherty and Milner 1999; Henny and Jones 2008; Ludkiewicz et al. 2002, 2000; Leranth and Frotscher 1987), suggesting that the spatial arrangement of ACh release sites varies across cell types and between regions. In contrast, volumetric transmission could be defined as a random pattern of release sites that release ACh onto nonspecific postsynaptic
targets. Therefore, by examining the spatial distribution of cholinergic boutons, principles could emerge that confer cell-type specificity to cholinergic transmission. Towards this end, we applied the modified Ripley’s K statistic to examine the spatial distribution of cholinergic boutons in the vicinity of hippocampal neurons. We find that this statistic can reveal targeting of boutons to the immediate vicinity of a neuron, distinguishing the empirical spatial distribution of these boutons from a random distribution.

5.2 Confocal imaging and image processing in Image J

Inhibitory interneurons in the hippocampus were visualized with GAD65-GFP mice (López-Bendito et al. 2004), a transgenic mouse line that labels distinct subpopulations of inhibitory interneurons with green fluorescent protein (GFP) (Wierenga et al. 2010). Cholinergic terminals were visualized with an antibody to the vesicular acetylcholine transporter (vAChT), similarly to previously described (Cea-del Rio et al. 2010). GAD65 GFP-positive neurons and vAChT-positive terminals were sequentially imaged using an Olympus Fluoview 1000 confocal microscope. All analyzed images were gathered using a 60x objective at 2x magnification. For the image stack in Figure 13A, ninety-three slices of 1024x1024 images were gathered at 0.5 micron z-step size. Images were exported to TIFF image formats, deconvolved with Huygens Essential Software (Scientific Volume Imaging Co., The Netherlands) and further processed using ImageJ software. To eliminate subjectivity and to maintain a uniformity of methodology across images, images first were thresholded using an automated Otsu threshold algorithm in both red and green channels, reducing the image data to binary. This image segmentation process limited the inclusion of background fluorescence in subsequent processing. Thresholded images were then counted at a minimum voxel size using the ImageJ plugin ObjectCounter3D. The minimum voxel size was calculated from a vAChT bouton size of 0.5 microns in diameter, which was based on empirical measures and available literature (Wouterlood et al. 2007, 2008). For the confocal stack in Figure 13A, 9737 objects (or “events”) were detected based on a minimum object size of 128 voxels. The x, y, and z coordinates of each of the 9737 objects were exported from ObjectCounter3D as a text file. The somatodendritic domain of the GAD65-GFP+ neuron was isolated by setting the object size to a minimum of 2,500 voxels, which excluded smaller objects in the green channel. The isolated neuronal body (green channel) was then exported to MATLAB as a TIFF file.

5.3 Image processing within MATLAB and R

From this revised TIFF file, MATLAB was used to find the surface points of the neuron. Three text files containing image data were created, two of which were the neuron surface points and the object locations both in ordered triplets. The third is a 3-dimensional binary array indicating the voxels that intersect the neuron volume. (This text file is a vector where location within the vector preserves the 3-dimensional array indices.) The
text files were then uploaded into the statistical program R and the Modified K function was implemented. Text files are provided with the supplementary materials.

In this application, the neuron is the mass and the 9737 objects are the events. The analysis shown in Figure 14 is from a single neuron. Function outputs are graphed similar to the output of Ripley’s K and L functions. Inference is taken from the more informative Modified L function in Figure 14. The Modified L function illustrates that the objects are significantly clustered about the neuron surface at small distances (0-5 microns).

![Diagram of neuron and objects](image)

**Figure 13**: This neuron in particular was imaged with 93 slices separated by 0.5\(\mu\)m. Each slice is then altered for clarity and then split in order to distinguish the neuron from objects (events). Locations of the neuron surface and objects are then taken from the respective image.
6 Discussion

The Modified K function and its application to neuroscience is a novel and useful method for determining aggregation about a mass. The simulation study included proves that the Modified K function operates well when a mass is present in the study region. The simulation study also increases knowledge of how the $K_M$ reacts to different kinds of spatial dependence upon a mass. If the Modified K function is used, the simulation study should be utilized to increase insight in how to interpret results.

The application to the Modified K function has demonstrated feasibility of the approach, that that there is clustering of ACh release points about a mass. Therefore, consistent with previous anatomical studies (Dougherty and Milner 1999; Henny and Jones 2008; Ludkiewicz et al. 2002, 2000; Leranth and Frotscher 1987), we have shown that the connectivity between cholinergic neurons and their cellular targets is non-random, that the cholinergic fibers preferentially make contact with the neuron. The natural progression from this study would be to look at multiple neurons and compare neurons that exhibit aggregation and randomness. This raises several points. First, although possible to detect in our simulations, we have not demonstrated whether neurons exist in which events (ACh release sites) are repulsed from the neuronal mass. We are planning to explore this question in a future study. Secondly, as the volume of study area is scaled up, there are inevitable complications that occur with multiple neurons in the study area. This is not studied in this paper, but the Modified K function could be adjusted for such study regions. Events would need to be assigned to a single mass or multiple masses.
depending on the strength of event to mass spatial dependence. This consideration of multiple masses is of importance, as existing images within our data set have multiple neurons within the study region. Future work could further explore practical limits to the study area and find an ideal study region size. As of now the ideal study region size is one that contains a single neuron.

In the future, we would like to incorporate this method into existing computational frameworks, such as the RipleyGUI environment (Hansson et al. 2013). In addition, there are opportunities to increase computational efficiency, either by streamlining or compiling the code or by making the code compatible with supercomputing resources.

There are assumptions in our approach that are clearly oversimplifications. For example, we assume that the release sites are point sources. Future work could incorporate complex structures to examine relative to a neuron. Finally, future work could incorporate diffusion of neurotransmitter from the release sites under different conditions (i.e. under circumstances in which ACh degradation has been blocked).

In conclusion, as spatial data within neuroscience are becoming more prevalent, our new methodology will provide an adequate tool for addressing spatial relationships and will uncover more in depth information when point processes become more complex.

Supplementary Materials

**Data:** The three data files described in section 5.5 are included in the zipped file DataZip.zip. (Cell 1_1-5 event.txt, Cell 1_1-5 mass.txt, Cell 1_1-5 cell.txt)

**R Code:** Two R files are included to accompany chapters 4 and 5. These files will reproduce most figures found in these chapters. Chapter 4 code will not reproduce simulations without additional coding, but should be used to explore how Modified L results react toward points patterns.(supplement_chapter4.R, supplement_chapter5.R)

**R Functions:** Three R functions needed to run R files.(ModKFun.R, plotK.R, ModKFunGAD.R)

References


