Research Article

Statistics in Medicine

Received 20 June 2011, Accepted 19 October 2012 Published online 22 November 2012 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.5675

Fast subset scan for multivariate event detection

Daniel B. Neill,[a*†] Edward McFowland III[a] and Huanian Zheng[b]

We present new subset scan methods for multivariate event detection in massive space–time datasets. We extend the recently proposed 'fast subset scan' framework from univariate to multivariate data, enabling computationally efficient detection of irregular space–time clusters even when the numbers of spatial locations and data streams are large. For two variants of the multivariate subset scan, we demonstrate that the scan statistic can be efficiently optimized over proximity-constrained subsets of locations and over all subsets of the monitored data streams, enabling timely detection of emerging events and accurate characterization of the affected locations and streams. Using our new fast search algorithms, we perform an empirical comparison of the Subset Aggregation and Kulldorff multivariate subset scans on synthetic data and real-world disease surveillance tasks, demonstrating tradeoffs between the detection and characterization performance of the two methods. Copyright © 2012 John Wiley & Sons, Ltd.

Keywords: algorithms; disease surveillance; event detection; scan statistics; spatial scan

1. Introduction

This work develops new subset scan methods and fast search algorithms for accurate and computationally efficient event detection in multivariate space–time data. Event detection is an important tool for public health, law enforcement, and other agencies responsible for the public good, enabling them to respond rapidly to potential threats including disease outbreaks, terrorist attacks, and natural disasters. In many of these applications, both early detection and accurate characterization of events are essential. For example, in disease surveillance, we wish to know whether an outbreak is occurring, what type of outbreak is present, and which areas have been infected, thus enabling a timely and effective public health response. Numerous methods have been developed for event detection in spatial and space–time data, including the spatial scan statistic [1] and many recently proposed variants and extensions. Typical spatial scan methods maximize a likelihood ratio statistic over a large set of space–time regions, identifying anomalous clusters that may correspond to emerging events.

In many applications, the timeliness and accuracy of event detection can be dramatically improved by integrating information from multiple data streams. In disease surveillance, early indicators of an emerging outbreak include data from hospital emergency departments, over-the-counter medication sales, school and work absenteeism, environmental sensors, and many others [2]. By combining data from multiple sources, or multiple streams of data from a single source (e.g., case counts for different disease symptoms), event detection methods can detect increasingly subtle signals that occur earlier in the progression of an outbreak [3]. Many recent variants of the spatial and space–time scan statistics have been proposed to incorporate multiple streams [4–7], but little work has been done to compare the effectiveness of these methods or to enable them to scale up to the increasingly large amounts of data available for detection tasks.

[a] *Event and Pattern Detection Laboratory, H.J. Heinz III College, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A.*

[b] *Department of E-commercial Product Marketing, Baidu Inc., Beijing 100085, China*

[*] *Correspondence to: Daniel B. Neill, H.J. Heinz III College, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A.*

[†] *E-mail: neill@cs.cmu.edu*

In this work, we present efficient search algorithms that enable us to scale up multivariate event surveillance to massive datasets in two distinct ways. First, our methods can be used to search over all proximity-constrained subsets of locations (thus including not only circular regions but also irregularly shaped regions) even when the number of locations is large. Previous work by Patil and Taillie [8], Duczmal and Assuncao [9], and others demonstrate that searching over irregular regions can dramatically improve the timeliness of event detection and can much more precisely pinpoint the spatial region affected by an event. Second, our methods can efficiently integrate information from many data streams, thus both enhancing detection power and providing more complete situational awareness by accurately *characterizing* the emerging events. Whereas complete characterization of an event such as a disease outbreak typically requires follow-up investigation by a domain expert, our methods can accurately identify the affected locations and the affected subset of the monitored data streams. As noted by Rolka *et al.* [10], 'the growing availability of data streams for biosurveillance requires corresponding growth in methodologies to analyze them', and the ability to integrate information from many data streams has become increasingly important for the creation of surveillance systems, such as HealthMap [11], which combine traditional health data with a continuously growing number of electronic media sources such as news reports, alert services, and patient self-reports.

In the multivariate event detection setting, we wish to optimize some measure of 'anomalousness' or 'interestingness', such as a spatial scan statistic, over subsets of the monitored locations and data streams, thus enabling us to detect emerging events, to characterize the affected streams, and to pinpoint the affected spatial region. This optimization task presents serious computational challenges: an exhaustive search over all subsets of the data is computationally infeasible, scaling exponentially with the numbers of locations and streams. Typical spatial scan methods either restrict the search space (e.g., searching over the much smaller set of circular regions) or perform an approximate heuristic search, resulting in reduced detection power and lower accuracy. However, Neill [12] demonstrated that many commonly used spatial and space–time scan statistics, including Kulldorff's original spatial scan [1], satisfy a property ('linear-time subset scanning' or LTSS) that allows extremely efficient unconstrained optimization over all subsets of spatial locations. For scan statistics satisfying the LTSS property, we can find the optimal subset of locations by ordering the locations according to some 'priority' function and searching over groups consisting of the top-j highest-priority locations, requiring only a linear rather than exponential number of subsets to be evaluated. Spatial proximity constraints are also incorporated by performing separate efficient searches over the local neighborhood of each spatial location. However, extension of this 'fast subset scan' framework from univariate to multivariate detection is non-trivial, and different algorithmic approaches are necessary depending on whether we assume constant or independent risks across the monitored streams.

In the remainder of this paper, we present new fast search algorithms for the multivariate spatial and space–time scan statistics, enabling computationally efficient detection of irregularly shaped space–time clusters even when the numbers of spatial locations and data streams are large. We use these efficient algorithms to perform a detailed empirical comparison of two variants of the multivariate subset scan. One of these methods, which we call Subset Aggregation, is an extension of previous work by Burkom [4], whereas the other approach was previously proposed by Kulldorff *et al.* [5]. We describe the Subset Aggregation and Kulldorff methods in Section 2 and present fast, scalable algorithms for both methods in Section 3. In Sections 4 and 5, we compare the run time and accuracy of naive and fast algorithms for the Subset Aggregation and Kulldorff methods, respectively. Sections 6 and 7 present a detailed empirical comparison of the two methods on synthetic data and on real-world disease outbreak detection tasks, respectively, demonstrating the inherent tradeoffs between detection power and characterization accuracy. For both methods, our experiments show that searching over proximity-constrained subsets of locations substantially improves detection power and spatial accuracy as compared with the traditional circular scan approach used in [1, 4, 5] and that our fast subset scan algorithms make this search over subsets computationally feasible for massive, multivariate datasets.

2. Multivariate event detection

In the multivariate event detection problem considered here, we monitor a set of data streams $\{d_1 \ldots d_M\}$ over time at a common set of spatial locations $\{s_1 \ldots s_N\}$, with the goal of rapidly and accurately detecting emerging patterns. For each data stream d_m and location s_i, we are given a time series of observed real-valued counts $c_{i,m}^t$. For example, in *disease surveillance*, we monitor multiple sources of electronically available public health data, such as hospital visits and medication sales, in order to detect emerging outbreaks of disease. For this application, each count $c_{i,m}^t$ could represent the number of observed disease cases in a specific syndrome category d_m, for a specific zip code s_i, on a given day. For each stream d_m and location s_i, we first compute the time series of expected counts (or 'baselines') $b_{i,m}^t$ using the historical data for that stream and location [13] and then compare actual and expected counts. Here we assume that a simple, 28-day moving average is used to compute baselines. We could also easily incorporate other methods of estimating the expected distribution of counts, such as using census populations or catchment areas, into our fast subset scan framework. We wish to detect any *spatial region* (set of nearby locations) where the recent counts for some subset of the monitored data streams are significantly higher than expected. In disease surveillance, this corresponds to an abnormally high incidence of disease cases in an area, which may indicate an emerging outbreak.

The *spatial and space–time scan statistics* are commonly used methods for event detection [1, 14]. They are in wide use for monitoring health data, with the goal of detecting clusters of disease cases due to chronic environmental exposures [15, 16], infectious disease outbreaks [17], or bioterrorist attacks [18]. These methods maximize a score function $F(S)$ over a large set of spatial regions S, each consisting of some subset of locations s_i. Typical spatial scan methods constrain the size and shape of the spatial region S and perform an exhaustive search over all regions satisfying the given constraints. Kulldorff's original method [1] assumed circular, purely spatial search regions, but recent variants search for elongated [19, 20] or irregular shapes [8, 9, 21] and scan over time as an additional search dimension [22, 23]. Finally, an estimate of statistical significance for each region is computed by randomization testing, and any significant regions are reported to the user. Neill *et al.* [13] developed an *expectation-based* scan statistic that first computes the expected count $b_{i,m}^t$ corresponding to each observed count $c_{i,m}^t$ by time series analysis and then compares the actual and expected counts. This method adjusts for the spatial and temporal variability of the background data, significantly improving detection time.

Parametric scan statistics [1, 13, 18, 24] assume some parametric model (such as Poisson-distributed or Gaussian-distributed counts) and maximize the log-likelihood ratio statistic $F(S)$ over all regions S, where $F(S) = \log \frac{P(\text{Data} \mid H_1(S))}{P(\text{Data} \mid H_0)}$. The null hypothesis H_0 assumes no clusters (i.e., all counts are generated from the expected distribution), and the alternative hypothesis $H_1(S)$ assumes that counts in region S are increased by some multiplicative factor. Here we focus on the expectation-based Poisson (EBP) statistic [13], discussed in detail in subsequent text, but we note that our fast subset scan approaches can be used to efficiently optimize any log-likelihood ratio statistic $F(S)$ that satisfies the LTSS property [12]. The EBP statistic differs slightly from Kulldorff's original Poisson spatial scan statistic [1], which compares the ratios of count to baseline inside and outside region S. Neill [24] demonstrated that EBP has high detection power for both small and large affected regions, whereas Kulldorff's statistic has high detection power for small affected regions but low detection power for large affected regions [24].

Many other variants of the spatial and space–time scan statistics have been proposed, such as the nonparametric [6] and Bayesian [7, 25, 26] scan statistics. These variants have some advantages over the parametric approaches: nonparametric scan statistics may increase detection power in cases where parametric assumptions are violated, for example, when combining multiple disparate data streams, and Bayesian scan statistics can model and differentiate between multiple event types. Nevertheless, we focus here on the parametric case, comparing two methods for combining multiple data streams and demonstrating how we can scale up each method to massive, multivariate spatial and space–time datasets.

2.1. Multivariate scan statistics

We now review the derivation of the EBP scan statistic, as applied to the multivariate space–time setting. For the EBP statistic, we assume each count $c_{i,m}^t$ to be generated from a Poisson distribution with mean equal to the product of the expected count (or baseline) $b_{i,m}^t$ and an unknown 'relative risk' $q_{i,m}^t$. Under the null hypothesis H_0 of no events, we assume $q_{i,m}^t = 1$ everywhere. Under the alternative hypothesis $H_1(D, S, W)$, we assume $q_{i,m}^t > 1$ for a given subset of data streams $D \subseteq \{d_1 \ldots d_M\}$ in a given spatial region $S \subseteq \{s_1 \ldots s_N\}$ for the most recent W time steps $t = 0 \ldots W-1$, where $t = 0$ represents the current time step. We typically consider all subsets of data streams D, all spatial regions S satisfying some set of constraints (e.g., circular regions or proximity-constrained subsets of locations), and all time durations $W = 1 \ldots W_{\max}$ for some constant $W_{\max}$.

Given a subset (D, S, W), we define the following aggregate quantities, which will be used throughout the paper. For a given location $s_i \in S$ and stream $d_m \in D$, let $C_{i,m}$ denote the *aggregate count* $\sum_{t=0 \ldots W-1} c_{i,m}^t$, and let $B_{i,m}$ denote the *aggregate baseline* $\sum_{t=0 \ldots W-1} b_{i,m}^t$. We then define the aggregate counts $C_i = \sum_{d_m \in D} C_{i,m}$, $C_m = \sum_{s_i \in S} C_{i,m}$, and $C = \sum_{d_m \in D} \sum_{s_i \in S} C_{i,m}$. The aggregate baselines B_i, B_m, and B are defined analogously as sums of $B_{i,m}$. Finally, we define the EBP score function, $F_{\text{EBP}}(c, b) = c \log\left(\frac{c}{b}\right) + b - c$, if $c > b$, and 0 otherwise. In the univariate space–time setting, given a single data stream d_m, $F(D, S, W) = F_{\text{EBP}}(C_m, B_m)$ for the EBP statistic [13].

Here we consider two different methods for combining multiple data streams in the space–time setting. The first method, which we call Subset Aggregation, is an extension of an approach previously proposed by Burkom [4]. Burkom's approach is a simple, univariate aggregation of the multiple streams: the counts $c_{i,m}^t$ and baselines $b_{i,m}^t$ are aggregated across all of the monitored streams, and then the log-likelihood ratio score is computed from the aggregate count and baseline. Subset Aggregation extends this approach by maximizing the log-likelihood ratio statistic over all 2^M subsets of the M monitored data streams; for each subset of streams, we aggregate the counts and baselines for that subset of streams and compute the log-likelihood ratio score from the aggregate count and baseline. As shown subsequently, our Subset Aggregation method solves several of the problems inherent in extending Burkom's original approach: it is able to accurately characterize the affected subset of data streams and does not suffer from poor performance when the data sources are disparate in scale. The second method was previously proposed by Kulldorff *et al.* [5]. Kulldorff's multivariate scan statistic computes a separate log-likelihood ratio score for each data stream and then adds these scores across the monitored streams.

The Subset Aggregation and Kulldorff methods each make different assumptions on how the relative risks $q_{i,m}^t$ are affected by an event $H_1(D, S, W)$, resulting in a different log-likelihood ratio statistic $F(D, S, W) = \log \frac{P(\text{Data} \mid H_1(D,S,W))}{P(\text{Data} \mid H_0)}$ in each case. More precisely, Subset Aggregation assumes a constant relative risk $q_{i,m}^t = q$ across all affected locations, streams, and time steps, where the value of q is computed by maximum likelihood estimation. Kulldorff's method assumes a constant relative risk $q_{i,m}^t = q_m$ for each data stream d_m across all affected locations and time steps, where each stream's relative risk q_m is computed separately by maximum likelihood estimation. Both methods assume that the data streams are conditionally independent given the relative risks q_m. This assumption of conditional independence may not be true in practice but is commonly made for the univariate spatial scan (assuming conditional independence between locations), the space–time scan (assuming conditional independence between time steps), and the multivariate scan considered here. Given the relative risk q_m for each data stream d_m, we can write the EBP statistic as follows:

$$
\begin{aligned}
F_{\text{EBP}}(D, S, W \mid \{q_m\}) &= \log \prod_{d_m \in D} \prod_{s_i \in S} \prod_{t=0 \ldots W-1} \frac{\Pr\left(c_{i,m}^t \sim \text{Poisson}\left(q_m b_{i,m}^t\right)\right)}{\Pr\left(c_{i,m}^t \sim \text{Poisson}\left(b_{i,m}^t\right)\right)} \\
&= \sum_{d_m \in D} \left(C_m \log q_m + B_m(1 - q_m)\right)
\end{aligned} \tag{1}
$$

Typically, however, the values of the relative risks q_m are unknown. For Kulldorff's method, we assume data streams to be affected independently, and thus we assume a different relative risk q_m for each stream d_m. Given the maximum likelihood estimates $q_m = \max\left(1, \frac{C_m}{B_m}\right)$ for the EBP statistic, we obtain the following:

$$
\begin{aligned}
F_{\text{EBP}}^K(D, S, W) &= \sum_{d_m \in D} \max_{q_m > 1} \left(C_m \log q_m + B_m(1 - q_m)\right) \\
&= \sum_{d_m \in D} \left(C_m \log\left(\frac{C_m}{B_m}\right) + B_m - C_m\right) 1\{C_m > B_m\} = \sum_{d_m \in D} F_{\text{EBP}}(C_m, B_m)
\end{aligned} \tag{2}
$$

For Subset Aggregation, on the other hand, we assume all data streams to be affected to the same extent, and thus we assume a single relative risk q across all affected data streams $d_m \in D$. In this case, the maximum likelihood estimate of q for the EBP statistic is $q = \frac{C}{B}$:

$$\begin{aligned} F_{\mathrm{EBP}}^{SA}(D,S,W) &= \max_{q>1} \sum_{d_m \in D} (C_m \log q + B_m(1-q)) \\ &= \max_{q>1}(C \log q + B(1-q)) \\ &= \left(C \log\left(\frac{C}{B}\right) + B - C\right) 1\{C > B\} = F_{\mathrm{EBP}}(C,B) \end{aligned} \tag{3}$$

Thus, for each subset of data streams D, we can reduce the Subset Aggregation scan statistic to a univariate space–time scan statistic applied to the aggregate counts and baselines for that subset of streams. However, we still consider Subset Aggregation (like Kulldorff's method) to be a *multivariate* subset scan method. It is the maximization of the score function $F(D,S,W)$ over subsets of streams D that makes this method multivariate and allows it to identify the affected subset of streams. This is exactly analogous to the spatial scan, which maximizes a score function $F(S)$ over spatial regions S: each individual computation of $F(S)$ is a function of the aggregate count and baseline for that region and does not take spatial information into account, but it is the maximization of $F(S)$ over multiple spatial regions that makes it a 'spatial' scan.

3. Accelerating multivariate event detection

In the multivariate event detection problem considered here, our primary goal is to find the most anomalous space–time regions by maximizing the score function $F(D,S,W)$ over subsets of data streams $D \subseteq \{d_1 \ldots d_M\}$, subsets of spatial locations $S \subseteq \{s_1 \ldots s_N\}$, and time durations $W = 1 \ldots W_{\max}$. Identification of the highest-scoring subsets (D,S,W) enables us to determine whether any significant clusters are present by randomization testing and to characterize each significant cluster by identifying the affected spatial region, time duration, and subset of data streams. In this work, we perform an unconstrained optimization over subsets of data streams, whereas our optimization over subsets of locations incorporates spatial constraints such as size, shape, and proximity.

However, this problem formulation creates two serious computational challenges. First, because there are exponentially many subsets of streams to consider, $O(2^M)$ for a multivariate dataset with M streams, an exhaustive search over all subsets of streams is computationally infeasible when the number of streams is large. Second, because there are exponentially many subsets of spatial locations, $O(2^N)$ for a spatial dataset with N locations, an unconstrained search over subsets of locations is typically infeasible. As discussed in [12], typical spatial scan methods either reduce the search space, considering only a polynomial number of subsets, or perform a heuristic search. For example, Kulldorff's original spatial scan [1] searches over only the $O(N^2)$ distinct circular regions centered at a location; other methods search over rectangles [19], ellipses [20], or cylinders [23]. Although such approaches reduce computational complexity, detection power tends to be low for patterns that do not correspond well to the subsets being searched. For example, a search over circles has high power to detect compact clusters but low power to detect elongated or irregular clusters. Heuristic search methods include [9], which uses simulated annealing to search over the space of all connected clusters, and [27], which uses a genetic algorithm to maximize a penalized likelihood ratio statistic. The disadvantage of these heuristic search methods is that they are not guaranteed to find a subset that is optimal (maximizes the score function) or even close to optimal. Other recently proposed methods for computationally efficient event detection, such as [19] and [28], efficiently and optimally search over the space of $O(N^4)$ rectangular regions. However, neither of these methods can integrate information from many data streams or search over all proximity-constrained subsets of locations, as in the present work.

Thus, we develop new algorithms that find the highest-scoring subsets of locations and data streams *without* an exhaustive search. As we will show, these approaches enable efficient optimization of the EBP scan statistic for multivariate space–time data, even when the numbers of locations and streams are large. While we first consider the case of unconstrained optimization over subsets of locations and streams, we can easily incorporate spatial proximity constraints into our fast subset scan framework, and such constraints are essential to avoid detecting spatially dispersed sets of locations that would not be considered as 'clusters'. Thus, our results focus on the 'fast localized scan' described in the following text, incorporating spatial proximity constraints defined by the local neighborhood of each spatial location rather than on unconstrained optimization.

In previous work [12], we defined the LTSS property and demonstrated that a large class of score functions satisfy this property, enabling each function to be efficiently optimized over the exponentially many subsets of the data without an exhaustive search. Intuitively, for a score function satisfying the LTSS property, we can find the highest-scoring subset of locations by first sorting the locations by priority and then considering only those subsets consisting of the top-j highest-priority locations. Formally, let $\{R_1 \dots R_N\}$ be a set of N data records, and let $F(S)$ be a set function mapping a subset of data records $S \subseteq \{R_1 \dots R_N\}$ to a real number. We refer to F as a 'score function' and $F(S)$ as the 'score' of subset S. Also, let $G(R_i)$ be a function mapping a single data record R_i to a real number. We refer to G as a 'priority function' and $G(R_i)$ as the 'priority' of data record R_i. Next, we define $R_{(j)}$, $j = 1 \dots N$, to be the jth highest-priority record, that is, the data record R_i with the jth highest value of $G(R_i)$. We can define the LTSS property as follows [12]:

Definition of LTSS
For a given set of records $\{R_1 \dots R_N\}$, the score function $F(S)$ and priority function $G(R_i)$ satisfy the *LTSS* property if and only if $\max_{S \subseteq \{R_1 \dots R_N\}} F(S) = \max_{j=1\dots N} F(\{R_{(1)} \dots R_{(j)}\})$.

If the LTSS property holds, we can efficiently maximize $F(S)$ over all subsets of the data by evaluating only N of the 2^N possible subsets. If the records $R_1 \dots R_N$ are already sorted by priority, we can maximize $F(S)$ in $O(N)$ time by stepping through the records in priority order and computing the score of each subset $S = \{R_{(1)} \dots R_{(j)}\}$. Otherwise, we must first sort the records by priority, which requires $O(N \log N)$ time. In this work, we will make use of the following theorem [12]:

Theorem 1
Let $F(S) = F(X, Y)$ be a function of two additive sufficient statistics of subset S, $X(S) = \sum_{R_i \in S} x_i$ and $Y(S) = \sum_{R_i \in S} y_i$, where x_i and y_i depend only on record R_i. Assume that $F(S)$ is monotonically increasing with $X(S)$, that all y_i values are positive, and that $F(X, Y)$ is convex. Then $F(S)$ satisfies LTSS with priority function $G(R_i) = \frac{x_i}{y_i}$.

As shown in [12], the function $F_{\text{EBP}}(c, b)$ (defined in §2.1) is a convex function of c and b and is monotonically increasing with c. It follows from Theorem 1 that this statistic satisfies the LTSS property, enabling efficient maximization of the score function over all subsets of locations. In the univariate space–time setting, given a single data stream d_m, we must compute $\max_{S \subseteq \{s_1 \dots s_N\}} F(D, S, W)$ separately for each temporal window size $W = 1 \dots W_{\max}$. For a given W, we can express $F_{\text{EBP}}(D, S, W)$ as $F_{\text{EBP}}(C_m, B_m) = F_{\text{EBP}}\left(\sum_{s_i \in S} C_{i,m}, \sum_{s_i \in S} B_{i,m}\right)$, and thus F_{EBP} satisfies LTSS with priority function $G(s_i) = \frac{C_{i,m}}{B_{i,m}}$. We can sort the locations by priority and then search over subsets consisting of the j highest-priority locations for $j = 1 \dots N$. The highest-scoring subset of locations, $\arg\max_S F(D, S, W)$, is guaranteed to be among the subsets searched, thus reducing the computational complexity from $O(2^N)$ to $O(N \log N)$. In the next two subsections, we consider how we can extend LTSS to the Subset Aggregation and Kulldorff multivariate subset scans, respectively. Each of these two methods has different properties, thus requiring the development of substantially different algorithms to make them computationally efficient and scalable for massive datasets. Although our discussion will focus on the EBP statistic, we note that we can easily extend these algorithms to other score functions satisfying the LTSS property, including the expectation-based Gaussian and exponential scan statistics [12].

3.1. *Three fast algorithms for the subset aggregation scan statistic*

As shown in §2.1, we can express the Subset Aggregation scan statistic as $F_{\text{EBP}}^{SA}(D, S, W) = F_{\text{EBP}}(C, B)$ for the EBP statistic, where C and B are the count and baseline aggregated over all data streams $d_m \in D$, all spatial locations $s_i \in S$, and all time steps $t = 0 \dots W - 1$. For Subset Aggregation, optimizing over subsets of locations and optimizing over subsets of streams are both computationally challenging. A naive approach would search exhaustively over all such subsets for each temporal window $W = 1 \dots W_{\max}$, resulting in a total complexity of $O(W_{\max} 2^{N+M})$ for a dataset with N locations and M streams. We denote this approach by NN, as it is 'naive' with respect to searching over subsets of locations and subsets of streams.

However, we can easily accelerate the Subset Aggregation scan statistic using the LTSS property defined earlier. First, for a fixed set of data streams $D \subseteq \{d_1 \dots d_M\}$ and a fixed temporal window W, we can use LTSS to efficiently optimize over all subsets of locations. To see this, we rewrite the EBP statistic as $F_{\mathrm{EBP}}(C, B) = F_{\mathrm{EBP}}\left(\sum_{s_i \in S} C_i, \sum_{s_i \in S} B_i\right)$. This statistic satisfies LTSS, with priority function $G(s_i) = \frac{C_i}{B_i}$. We first compute the aggregates over all streams $d_m \in D$ and all time steps $t = 0 \ldots W - 1$ for each location $s_i \in S$. We then compute the priority of each location, sort the locations by priority, and search over subsets consisting of the j highest-priority locations for each $j = 1 \ldots N$. For the given data streams D and temporal window W, the highest-scoring subset of locations $\arg\max_S F(D, S, W)$ is guaranteed to be one of these N subsets, thus reducing the complexity from $O(2^N)$ to $O(N \log N)$.

Similarly, for a fixed set of spatial locations $S \subseteq \{s_1 \ldots s_N\}$ and a fixed temporal window W, we can use LTSS to efficiently optimize over all subsets of data streams. In this case, we rewrite the EBP statistic as $F_{\mathrm{EBP}}(C, B) = F_{\mathrm{EBP}}\left(\sum_{d_m \in D} C_m, \sum_{d_m \in D} B_m\right)$. This statistic satisfies LTSS, with priority function $G(d_m) = \frac{C_m}{B_m}$. We first compute the aggregates over all locations $s_i \in S$ and all time steps $t = 0 \ldots W - 1$ for each data stream $d_m \in D$. We then compute the priority of each stream, sort the streams by priority, and search over subsets consisting of the j highest-priority streams for each $j = 1 \ldots M$. For the given spatial region S and temporal window W, the highest-scoring subset of streams $\arg\max_D F(D, S, W)$ is guaranteed to be one of these M subsets, thus reducing the computational complexity from $O(2^M)$ to $O(M \log M)$.

Given these two efficient optimization steps as building blocks, we now consider how to jointly maximize $F(D, S, W)$ over all subsets of streams D, subsets of locations S, and temporal windows W. First, if the number of data streams is small, we can exhaustively search over all 2^M subsets of streams for each temporal window size $W = 1 \ldots W_{\max}$. For each subset of streams and each temporal window, we efficiently optimize over all subsets of locations, as described earlier. This algorithm, which we denote by FN (for fast optimization over subsets of locations and naive optimization over subsets of streams), has a computational complexity of $O(W_{\max} 2^M N \log N)$. Similarly, if the number of spatial locations is small, we can exhaustively search over all 2^N subsets of locations for each temporal window size $W = 1 \ldots W_{\max}$. For each subset of locations and each temporal window, we efficiently optimize over all subsets of streams, as described earlier. This algorithm, which we denote by NF (for naive optimization over subsets of locations and fast optimization over subsets of streams), has a computational complexity of $O(W_{\max} 2^N M \log M)$. If we consider only a smaller number of spatial regions, for example, searching over the $O(N^2)$ circular regions rather than searching over all $O(2^N)$ subsets of locations, then the complexity of NF is reduced to $O(W_{\max} N_S M \log M)$, where N_S is the total number of regions searched.

If the numbers of locations and streams are both large, we propose a third algorithm, which we denote by FF (for fast optimization over subsets of locations and subsets of streams). For each temporal window size $W = 1 \ldots W_{\max}$, we begin by randomly choosing a subset of streams $D \subseteq \{d_1 \ldots d_M\}$. We choose a value p uniformly at random between 0 and 1 and independently include each stream d_m with probability p. We then iterate between two steps: efficiently optimizing over all subsets of locations for the current subset of streams and efficiently optimizing over all subsets of streams for the current subset of locations. In the first step, we set $S = \arg\max_{S \subseteq \{s_1 \ldots s_N\}} F(D, S, W)$ for the given D and W, and in the second step, we set $D = \arg\max_{D \subseteq \{d_1 \ldots d_M\}} F(D, S, W)$ for the given S and W. Both steps are guaranteed not to decrease the score $F(D, S, W)$, and the FF algorithm iterates between these two steps until it converges to a local maximum of the score function. At the local maximum, the subset of locations is conditionally optimal given the subset of streams, and the subset of streams is conditionally optimal given the subset of locations. However, unlike the NN, FN, and NF algorithms, this iterative ascent procedure does not guarantee convergence to the global maximum of the score function $F(D, S, W)$. Thus, FF performs multiple random restarts using different initial subsets of streams. Our experiments demonstrate that this procedure will converge to a near-optimal score with high probability. For a dataset with N locations and M streams, the resulting algorithm has a complexity of $O(RZW_{\max}(NM + N \log N + M \log M))$, where R is the number of random restarts and Z is the average number of iterations required for convergence. In this expression, the $O(NM)$ term results from aggregating the counts and baselines over locations and streams, whereas the $O(N \log N)$ and $O(M \log M)$ terms result from sorting the locations and streams by priority, respectively.

We note that earlier efforts to find clusters using multiple data sources [4, 5] employed searches only over the space of circular regions rather than all proximity-constrained subsets of locations. Like FN, this method scales efficiently with the number of spatial locations, but we show in the following text that the limitation of the scan to circles results in reduced detection power for elongated or irregular clusters. Additionally, we can use the NF method presented here to efficiently search over subsets of the monitored data streams for each circular region.

3.2. *A fast algorithm for Kulldorff's multivariate scan statistic*

As shown in §2.1, we can express Kulldorff's multivariate scan statistic as $F^K_{\text{EBP}}(D, S, W) = \sum_{d_m \in D} F_{\text{EBP}}(C_m, B_m)$ for the EBP statistic, where C_m and B_m are the count and baseline for stream d_m, aggregated over all spatial locations $s_i \in S$ and all time steps $t = 0 \ldots W - 1$. For Kulldorff's method, optimization of $F(D, S, W)$ over all subsets of data streams $D \subseteq \{d_1 \ldots d_M\}$ is computationally trivial: because the function F_{EBP} is non-negative, we need only compute $F(D, S, W)$ for the single subset containing all M data streams, $D = \{d_1 \ldots d_M\}$. We can then find the minimal subset of streams $D^* = \arg\max_D F(D, S, W)$ by excluding any streams with zero scores: we can exclude stream d_m when $C_m \leqslant B_m$. Thus, we can define the naive Kulldorff method (NK) as an exhaustive search over spatial regions S for each temporal window $W = 1 \ldots W_{\max}$, performing the preceding (trivial) optimization of $F(D, S, W)$ over subsets of data streams for each combination of S and W. The computational complexity of this naive approach is $O(W_{\max} M N_S)$, where M is the number of data streams and N_S is the number of spatial regions considered. However, if we perform an unconstrained search over subsets of N locations, there are $O(2^N)$ regions to consider, giving a total complexity of $O(W_{\max} M 2^N)$. Reducing the complexity from exponential to linear in the number of locations is challenging because, although the score of each of the M data streams is a convex function of the aggregate count and baseline for that stream (and thus satisfies the LTSS property), the sum of these M scores cannot be expressed as a function of two additive sufficient statistics and thus is not guaranteed to satisfy LTSS.

Our solution is to condition on the relative risk q_m for each data stream d_m. As shown in §2.1, we can write $F_{\text{EBP}}(D, S, W \mid \{q_m\}) = \sum_{d_m \in D} (C_m \log q_m + B_m(1 - q_m))$. For a given subset of streams D and a given temporal window size W, we can write $F_{\text{EBP}}(D, S, W \mid \{q_m\}) = \sum_{s_i \in S} G(s_i)$, where $G(s_i) = \sum_{d_m \in D} (C_{i,m} \log q_m + B_{i,m}(1 - q_m))$. Thus, for given values of $q_1 \ldots q_M$ and for a given temporal window W, we can easily compute the highest-scoring subset of locations $\arg\max_S F(D, S, W)$ by including all locations s_i for which $G(s_i)$ is positive. We can also easily compute the maximum likelihood values of $q_1 \ldots q_M$ for a given subset of locations S and temporal window W: we set $q_m = \frac{C_m}{B_m}$ for each data stream d_m.

Thus, we propose the following 'fast Kulldorff' (FK) algorithm. For each temporal window size $W = 1 \ldots W_{\max}$, we begin by randomly initializing the values of the relative risks q_m, $m = 1 \ldots M$. We choose a value p uniformly at random between 0 and 1 and independently include each stream with probability p. For included streams, q_m was set to $\exp(x_m)$, where $x_m \sim \text{Uniform}[0, 2]$, and for excluded streams, q_m was set to 1. We then compute the subset of locations $S \subseteq \{s_1 \ldots s_N\}$ that maximizes $F(D, S, W \mid \{q_m\})$ and recompute the corresponding relative risks $q_1 \ldots q_M$ for region S by maximum likelihood estimation, as described earlier. The FK algorithm repeats these two steps, maximizing over subsets of locations for the current relative risk values and maximizing over relative risks for the current subset of locations, until it converges to a local maximum of the score function. At the local maximum, the subset of locations S is conditionally optimal given the relative risks $\{q_m\}$, and the relative risks are conditionally optimal given the subset of locations. However, unlike the NK algorithm, this iterative ascent procedure does not guarantee that a global maximum of the score function $F(D, S, W)$ will be obtained, and thus FK performs multiple random restarts with different initial values of the relative risks. As we discuss in the following text, our experiments demonstrate that this procedure will converge to a near-optimal score with high probability. We also note that, whereas a separate optimization of $F(D, S, W)$ must be performed for each temporal window size $W = 1 \ldots W_{\max}$, we need only to perform a single optimization for all data streams $D = \{d_1 \ldots d_M\}$ and then report all streams with positive scores, as described earlier. For a dataset with N locations and M streams, the resulting algorithm has a complexity of $O(RZW_{\max}NM)$, where R is the number of random restarts and Z is the average number of iterations required for convergence.

As for the Subset Aggregation method in the preceding text, we note that the version of the multivariate scan statistic originally proposed by Kulldorff *et al.* [5] searches only over the space of circular regions rather than all proximity-constrained subsets of locations. Like FK, this method scales efficiently with the number of spatial locations and the number of monitored data streams, but we show in the following text that the limitation of the scan to circles results in reduced detection power.

3.3. *Incorporating spatial constraints*

Whereas these results demonstrate the potential of LTSS to enable efficient unconstrained maximization of the score function for multivariate spatial and space–time data, we note that an unconstrained search over subsets is typically not sufficient to solve practical spatial detection problems, as it does not take the spatial proximity of locations into consideration and thus the highest-scoring 'region' may consist of a spatially dispersed set of locations. Optimizing over all 2^N subsets of locations without additional constraints may also increase the number of false positives (or equivalently, reduce detection power for a fixed false positive rate) because of the large number of hypotheses being tested. Our solution is the 'fast localized scan' approach first described in [12], in which we consider only subsets of the local neighborhoods consisting of a 'center location' s_i and its $k-1$ nearest neighbors, for a fixed constant neighborhood size k. Each of the N spatial locations must be separately considered as a possible center. Thus, a naive search algorithm (NN, NF, or NK) would require the evaluation of $N_S = O(N2^k)$ subsets of locations for each subset of streams D and each temporal window $W = 1 \dots W_{\max}$. This gives a total complexity of $O(W_{\max} N 2^{k+M})$ for NN, $O(W_{\max} N 2^k M \log M)$ for NF, and $O(W_{\max} N 2^k M)$ for NK, respectively.

However, the LTSS property allows us to efficiently maximize $F(D, S, W)$ for each local neighborhood by evaluating only $O(k)$ of the $O(2^k)$ subsets of locations. Assuming that the k locations are already sorted by priority, we need only to evaluate the subsets consisting of the j highest-priority locations, for $j = 1 \dots k$. This gives a total complexity of $O(W_{\max} 2^M (Nk + N \log N))$ for the FN algorithm. In this expression, the $O(N \log N)$ term results from sorting the locations by priority; the locations must be sorted only once for each temporal window W and each subset of streams D under consideration. For the FF algorithm, we have a total complexity of $O(RZW_{\max} N(kM + k \log k + M \log M))$, where R is the number of random restarts and Z is the average number of iterations per restart. For a given neighborhood of size k, the $O(kM)$ term results from aggregating counts and baselines, the $O(k \log k)$ term results from sorting locations by priority, and the $O(M \log M)$ term results from sorting streams by priority. Finally, for the FK algorithm, we have a total complexity of $O(RZW_{\max} NkM)$.

We note that the fast localized scan is very similar to the flexible spatial scan statistic (FlexScan) proposed by Tango and Takahashi [21], in that it searches over subsets of neighborhoods defined by a center location and its $k-1$ nearest neighbors. However, as we demonstrate in [12], the runtime of FlexScan scales exponentially with k, making it computationally infeasible for $k > 30$, whereas fast localized scan scales linearly with k, thus enabling efficient computation even when k is very large. Because FlexScan does not scale to the large numbers of locations and data streams described here, we do not include this method in our evaluation results presented subsequently. Another difference between the two methods is that fast localized scan can return a disconnected region if it satisfies the proximity constraints, whereas FlexScan requires the resulting region to be connected. Enforcing connectivity may be preferable in some problem settings, whereas in other cases proximity may be sufficient.

We also note the similarity between the univariate version of our fast subset scan approach and the upper level set (ULS) scan statistic proposed by Patil and Taillie [8], which has been widely applied to graph and network data. The ULS approach prioritizes the set of spatial locations by the ratio of count to baseline and considers the top-j highest-priority locations for each $j = 1 \dots N$. Rather than considering the subset consisting of all j locations, however, ULS enforces a connectivity constraint, considering the connected components of the subgraph formed by the top-j locations for each j. Thus, for a fully connected graph, ULS reduces to the unconstrained fast subset scan approach. However, fast subset scan is much more general than ULS: it is applicable for optimization of a large class of score functions, including both parametric and nonparametric scan statistics, and can incorporate a variety of constraints, such as spatial proximity or graph connectivity [12]. Most importantly, the LTSS property can be extended to enable efficient computation of multivariate space–time scan statistics, whereas ULS assumes univariate, purely spatial data. Finally, as shown in [12], ULS is not guaranteed to compute the highest-scoring connected cluster in the univariate setting, whereas fast subset scan can efficiently find the exact solution to constrained and unconstrained subset scan problems without an exhaustive search.

4. Comparison of fast and naive algorithms for subset aggregation

We now compare the run time and accuracy for the Subset Aggregation scan statistic using the four algorithms described earlier: NN (naive search over locations and streams), FN (fast search over locations and naive search over streams), NF (naive search over locations and fast search over streams), and FF (fast search over locations and streams). For each algorithm, we considered the EBP statistic with maximum temporal window size $W_{\max} = 1$. We compared the run time of the fast localized scan (searching over proximity-constrained subsets of locations and all subsets of streams) as a function of the neighborhood size k and the number of monitored data streams M. Each method was run on 647 days of emergency department (ED) data from $N = 97$ Allegheny County zip codes; this dataset is described in §7.1. Figures 1 and 2 compare the total run times of the FN, NF, and NN methods for the 647 days of data, performing a separate optimization over subsets of locations and streams for each day. As noted earlier, all three methods are guaranteed to find the highest-scoring subset of locations and streams. The run time of the NN method scaled exponentially with the neighborhood size k and number of data streams M. The run time of FN scaled linearly with k and exponentially with M, and the run time of NF scaled exponentially with k and linearly with M. Figure 1 compares the run times of FN and NN for $M = 8$ data streams, as a function of the neighborhood size k: FN required less than 0.73 s per day of data for all neighborhood sizes, whereas NN was computationally infeasible for all $k \geqslant 20$. Figure 2 compares the run times of NF and NN for a fixed neighborhood size $k = 10$, as a function of the number of monitored data streams M. For $k = 10$ and $M = 16$, NN required approximately 20 min per day of data, whereas FN required less than 0.65 s per day of data, a 1850× speedup.

The FN method enables fast, exact computation of the highest-scoring subset of locations and streams if the number of data streams is small, whereas the NF method enables efficient computation if the neighborhood size is small. If the neighborhood size and number of streams are both large, we must use the FF method. However, as noted earlier, FF converges to a conditional rather than global maximum of the score function $F(D, S, W)$ and thus is not guaranteed to find the highest-scoring subset of locations and streams. We define the *approximation ratio* as the largest value p such that the approximate method (FF) achieves a score within $(100 - p)\%$ of the global maximum score (computed by FN, NF, or NN) at

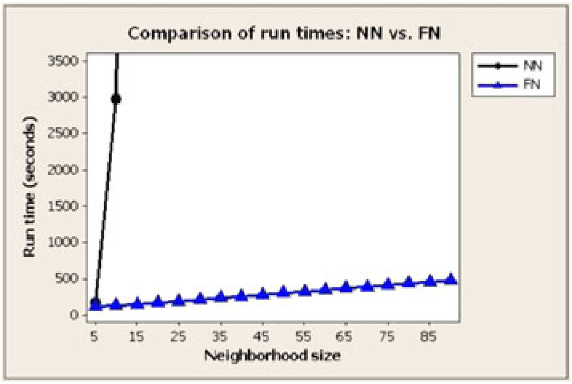

Figure 1. Comparison of run times of fast (FN) and naive (NN) Subset Aggregation algorithms for eight data streams for 647 days of emergency department data, as a function of the neighborhood size k.

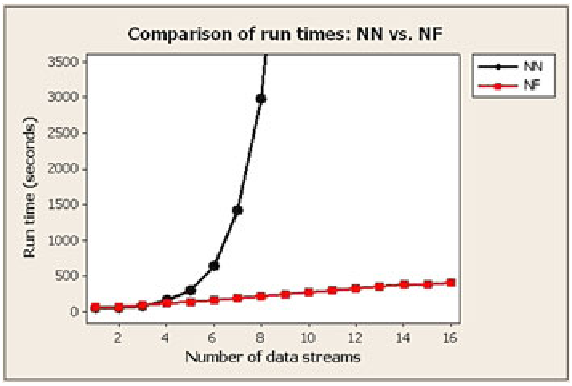

Figure 2. Comparison of run times of fast (NF) and naive (NN) Subset Aggregation algorithms with neighborhood size $k = 10$ for 647 days of emergency department data, as a function of the number of streams.

least p% of the time. For example, an approximation ratio of 95% would signify that FF achieves a score within 5% of the global maximum with 95% probability. We can easily compute the approximation ratio by first calculating the ratio of the maximum score computed by FF to the maximum score computed by FN, NF, or NN on each run, sorting these ratios, and stepping through the sorted list to find the maximum value of p.

Table I shows the speedup in run time and the approximation ratio of FF as compared with the three exact algorithms (FN, NF, and NN), as a function of the neighborhood size k and the number of monitored data streams M, using the 647 days of ED data and performing a separate optimization over subsets of locations and streams for each day. We used $R = 50$ random restarts for the FF algorithm. We computed the speedup by comparing FF with the fastest of the three exact algorithms for the given values of k and M. For $k = 1 \dots 7$, FF runs slower than NF, and for $M = 1 \dots 7$, FF runs slower than FN. Thus, we only present results for $k \geqslant 8$ and $M \geqslant 8$. As we can see from the table, FF demonstrates substantial speedups as compared with FN, NF, and NN when the neighborhood size k and number of streams M are both large. For $k = 15$ and $M = 16$ streams, the run time of FF was 0.36 s per day of data as compared with 13 s for NF. For $k = 90$ and $M = 16$, run time of FF was 0.82 s per day of data as compared with 2.7 min for FN. The approximation ratio of FF stayed relatively constant for varying k and M, decreasing slightly for large values of k. The approximation ratio remained above 98% for all cases tested, demonstrating that FF is able to find a near-optimal region with high probability. Finally, we found that the average number of iterations Z for the FF algorithm increased logarithmically with k and M, up to $Z = 5.45$ for $k = 90$ and $M = 16$.

5. Comparison of fast and naive algorithms for Kulldorff's method

We now compare the run time and accuracy of the fast (FK) and naive (NK) algorithms for Kulldorff's multivariate scan statistic. Figure 3 shows the run times of FK and NK for monitoring eight streams of ED data; total run times for 647 days of data are shown. As in the previous section, we considered the EBP statistic with a maximum temporal window size of $W_{\max} = 1$. We compared the run time of the fast

Table I. Comparison of run time and accuracy of FF algorithm, as compared with exact algorithms (FN, NF, and NN), for the Subset Aggregation scan statistic.

k	$M = 8$	$M = 10$	$M = 12$	$M = 14$	$M = 16$
8	1.1× / 99.7%	1.1× / 100%	1.1× / 99.8%	1.1× / 99.5%	1.1× / 99.7%
15	1.2× / 99.7%	2.2× / 99.8%	6.3× / 99.5%	20× / 99.4%	36× / 99.7%
30	1.3× / 99.3%	3.0× / 99.5%	8.9× / 99.1%	30× / 99.7%	112× / 99.2%
50	1.5× / 98.6%	3.8× / 98.7%	12× / 98.9%	41× / 98.6%	151× / 99.0%
70	1.6× / 98.5%	4.3× / 98.8%	14× / 99.0%	48× / 98.7%	179× / 98.2%
90	1.7× / 99.2%	4.8× / 98.9%	16× / 99.2%	53× / 98.9%	198× / 98.9%

Speedup and approximation ratio for FF (with $R = 50$ random restarts) as a function of the neighborhood size k and number of monitored data streams M.

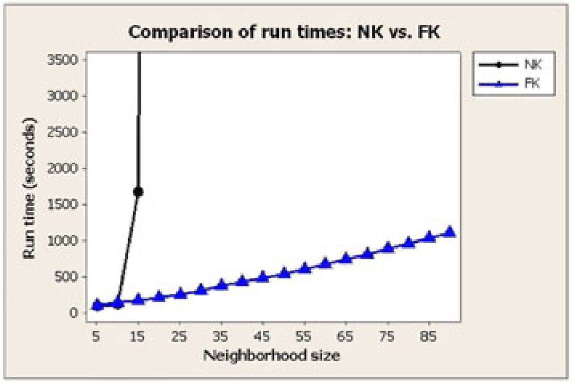

Figure 3. Comparison of run times of fast (FK) and naive (NK) algorithms for Kulldorff's multivariate scan statistic, monitoring eight data streams for 647 days of emergency department data, as a function of the neighborhood size k.

localized scan (searching over proximity-constrained subsets of locations and all subsets of streams) as a function of the neighborhood size k and the number of monitored data streams M. $R = 50$ random restarts were used for the FK algorithm. As expected, the run times of both NK and FK scaled linearly with the number of data streams M. However, as we can see from Figure 3, the run time of NK scaled exponentially with the neighborhood size k, whereas the run time of FK scaled linearly with k. For eight streams, the run time of NK was 97 s per day of data at $k = 20$, 28 h per day of data at $k = 30$, and approximately 3300 years per day of data at $k = 50$, whereas the run time of FK was less than 1.72 s per day of data for all neighborhood sizes.

As noted earlier, however, FK converges to a conditional rather than global maximum of the score function $F(D, S, W)$ and thus is not guaranteed to find the highest-scoring subsets of locations and data streams, whereas NK is guaranteed to find the global maximum. Table II shows the speedup in run time and the approximation ratio of FK as compared with NK, as a function of the neighborhood size k and the number of monitored data streams M. For $k = 1 \ldots 10$, FK runs slower than NK, whereas for $k > 20$, it is computationally infeasible to run NK for 647 days of data; thus, only neighborhood sizes between 11 and 20 are shown. FK demonstrated speedups of 9–15× at $k = 15$ and 265–529× at $k = 20$. The approximation ratio of FK remained approximately constant with varying neighborhood size and number of streams and remained above 93% for all cases tested, demonstrating that FK is able to find a near-optimal region with high probability. Finally, we found that the average number of iterations Z for the FK algorithm increased logarithmically with the neighborhood size k and number of streams M, up to a maximum of $Z = 4.55$ for $k = 90$ and $M = 16$.

6. Comparison of subset aggregation and Kulldorff methods on synthetic data

We now compare the power and spatial accuracy of the Subset Aggregation and Kulldorff multivariate subset scans on synthetic, purely spatial data, assuming $N = 256$ spatial locations mapped to a 16×16 grid. Our first set of simulations considered $M = 2$ monitored data streams, and our second set of simulations considered $M = 8$ monitored data streams for each spatial location. Synthetic datasets were created by drawing each count $c_{i,m}$ from a normal distribution with mean 100 and standard deviation 10 and setting each baseline $b_{i,m}$ to 100. Synthetic events were added to the data by incrementing the counts $c_{i,m}$ for 13 locations: a ‘center’ location s_i and all other locations with L_1 distance $d \leqslant 2$ from the center location. For a given data stream, each affected count was increased by an identical amount x_m. For our first set of experiments, with $M = 2$ data streams, we fixed $x_1 = 10$ and varied x_2 from 0 to 15. For our second set of experiments, with $M = 8$ data streams, we varied the number of ‘affected’ data streams between 1 and 8, setting $x_m = 8$ for affected streams and $x_m = 0$ for unaffected streams.

Finally, we conducted a third set of experiments with $M = 2$ to examine the effects of disparities in the *scale* of the data streams on the performance of each method. For example, in the public health domain, we might wish to monitor over-the-counter medication sales and ED visits, where the average daily count for over-the-counter medication sales would be much higher than the corresponding number of ED visits. For these experiments, we generated synthetic bivariate data where the first data stream had $b_{i,m} = 2500$ and $c_{i,m} \sim N(2500, 50)$, and the second data stream had $b_{i,m} = 100$ and $c_{i,m} \sim N(100, 10)$. We assumed that a signal was present only in the second stream, fixing $x_1 = 0$ and varying x_2 between 0 and 15. We hypothesized that a simple univariate aggregation of the two data streams would perform poorly, as the signal in the second stream would be overwhelmed by the noise in the first stream. Thus, we compared the performance of the multivariate scans with a univariate aggregation method, which monitors the sum of counts aggregated across the two monitored streams, as well as the performance of a univariate scan that only monitored the single affected stream.

Table II. Comparison of run time and accuracy of fast (FK) and naive (NK) algorithms for Kulldorff’s multivariate scan statistic.

k	$M = 1$	$M = 2$	$M = 4$	$M = 8$	$M = 16$
11	1.6× / 96.4%	1.4× / 93.2%	1.2× / 95.6%	1.1× / 95.5%	1.1× / 95.4%
15	15× / 97.1%	15× / 93.8%	11× / 95.2%	9.3× / 94.9%	9.0× / 95.7%
20	529× / 97.7%	507× / 94.5%	334× / 95.1%	283× / 95.5%	265× / 95.8%

Speedup and approximation ratio for FK (with $R = 50$ random restarts) as a function of the neighborhood size k and number of monitored data streams M.

For each of the preceding experiments, we generated 10,000 'background' datasets (with no injected counts) and 10,000 'inject' datasets. We then compared the methods using two criteria: detection power and spatial accuracy. We evaluated detection power by computing the proportion of inject datasets that had scores $F^* = \max F(D, S, W)$ higher than 95% of the 10,000 background datasets and thus would have been detected assuming an allowable false positive rate of 5%. We evaluated spatial accuracy by computing the average overlap coefficient between the true set of 13 affected locations S_T and the set of detected locations $S^* = \arg\max_S F(D, S, W)$: Overlap $= \frac{|S_T \cap S^*|}{|S_T \cup S^*|}$. For both the Subset Aggregation and Kulldorff methods, we used the EBP statistic and searched over the set of circular regions (assuming an L_1 distance metric on the grid). We note that the affected set of locations corresponded exactly to one of the circular regions being searched. Thus, we did not consider searching over subsets of locations but did search over subsets of streams by using the NK algorithm for Kulldorff's method and the NF algorithm for Subset Aggregation.

We show our results for the first set of experiments (assuming $M = 2$, $x_1 = 10$, and $x_2 = 0 \dots 15$) in Figure 4. As we can see from the figure, the detection power and spatial accuracy of the Subset Aggregation and Kulldorff methods were similar and improved with an increasing number of injected cases x_2. For most values of x_2, Kulldorff's method outperformed Subset Aggregation by a small amount with respect to both power and spatial accuracy: power was significantly improved ($p < 0.05$) for x_2 values between 3 and 8, and spatial accuracy was significantly improved ($p < 0.05$) for x_2 values between 3 and 6.

Thus, our results suggest that Kulldorff's method can improve detection power and spatial accuracy when the effects of an event vary across the different data streams. In Figure 5, we can see one possible reason for these differences in performance: Kulldorff's method detects both data streams much more frequently than Subset Aggregation. When x_2 is small, Subset Aggregation often fails to detect

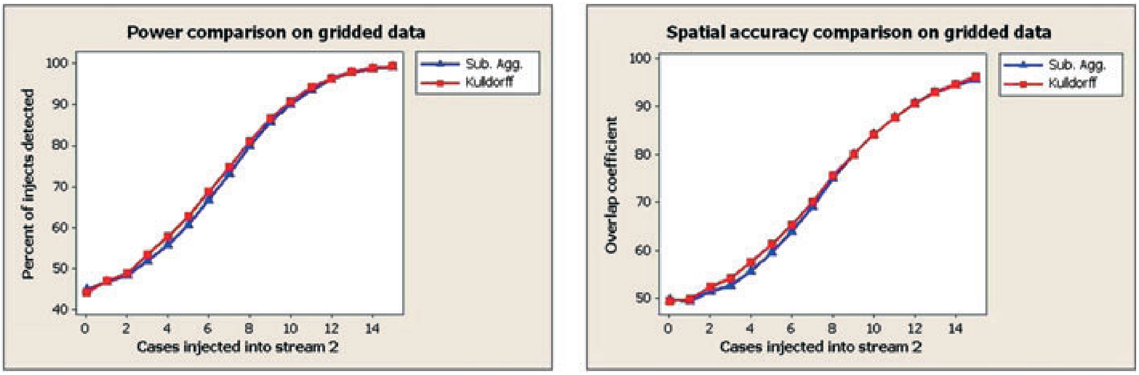

Figure 4. Comparison of detection power and spatial accuracy between Subset Aggregation and Kulldorff multivariate subset scans, assuming $M = 2$, $x_1 = 10$, and varying x_2. (a) Percent of injects detected, assuming a 5% false positive rate. (b) Average overlap coefficient between true and detected regions.

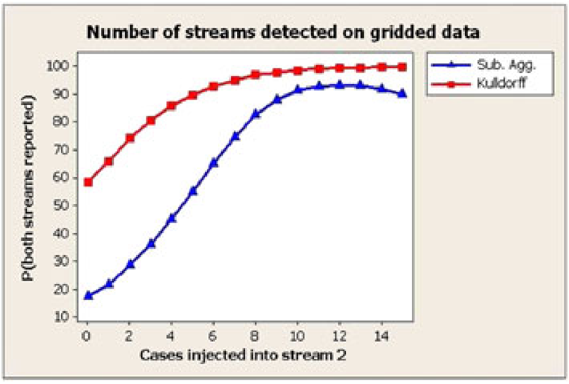

Figure 5. Percent of injects for which both of the two data streams are detected, for Subset Aggregation and Kulldorff multivariate subset scans, assuming $M = 2$, $x_1 = 10$, and varying x_2.

the weakly affected stream d_2, and when x_2 is large, it can fail to detect the less affected stream d_1. With $x_1 = 10$ and $x_2 = 0$, the Kulldorff and Subset Aggregation methods detect the affected stream 96% and 91% of the time, respectively; however, Kulldorff's method also reports the unaffected stream (incorrectly identifying that stream as 'affected') 63% of the time, as compared with 26% for Subset Aggregation. Similarly, for $x_1 = 10$ and $x_2 = 5$, the Kulldorff and Subset Aggregation methods detect the more affected stream 97% and 94% of the time, respectively; however, Kulldorff's method also detects the less affected stream 92% of the time, as compared with 61% for Subset Aggregation. In general, Kulldorff's method tends to detect all affected streams but often reports unaffected streams as well, whereas Subset Aggregation tends to detect only the most strongly affected streams if the inject size varies between streams.

Our results for the second set of experiments, in which only a subset of the eight monitored data streams were affected, also demonstrate substantial differences between the two methods. Subset Aggregation tended to detect the affected streams with high probability and report the unaffected streams with low probability, whereas Kulldorff's method tended to report both affected and unaffected streams, as shown in Figure 6. However, there were no significant differences in detection power or spatial accuracy between the Subset Aggregation and Kulldorff methods for this set of experiments.

We show our results for the third set of experiments, assuming two data streams of disparate scale, in Figure 7. As expected, a simple univariate aggregation of the monitored streams [4] is overwhelmed by the noise in the higher-count stream, resulting in low detection power and spatial accuracy. Whereas a univariate approach monitoring only the second stream demonstrated high detection power and accuracy, this approach would have no power to detect signals occurring in the first stream. However, both

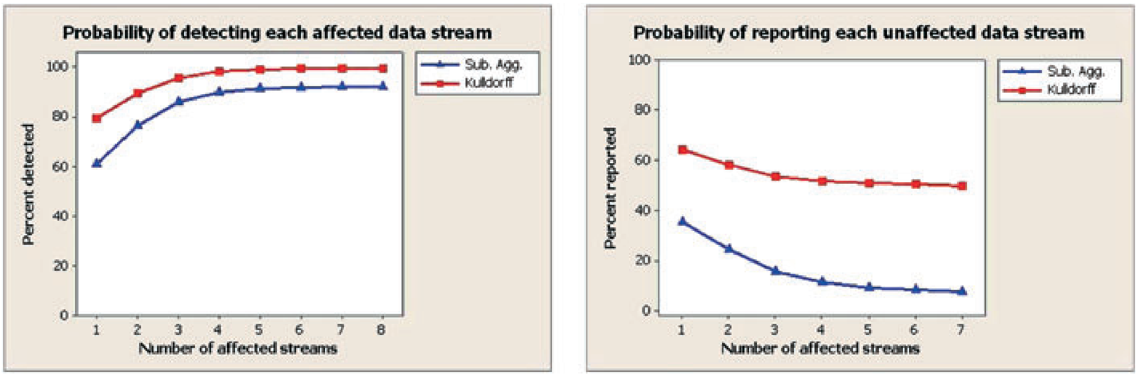

Figure 6. Comparison of stream detection probability between Subset Aggregation and Kulldorff multivariate subset scans, assuming $M = 8$, and one to eight affected streams with $x_m = 8$. (a) Probability (in percent) of detecting each affected stream. (b) Probability (in percent) of reporting each unaffected stream.

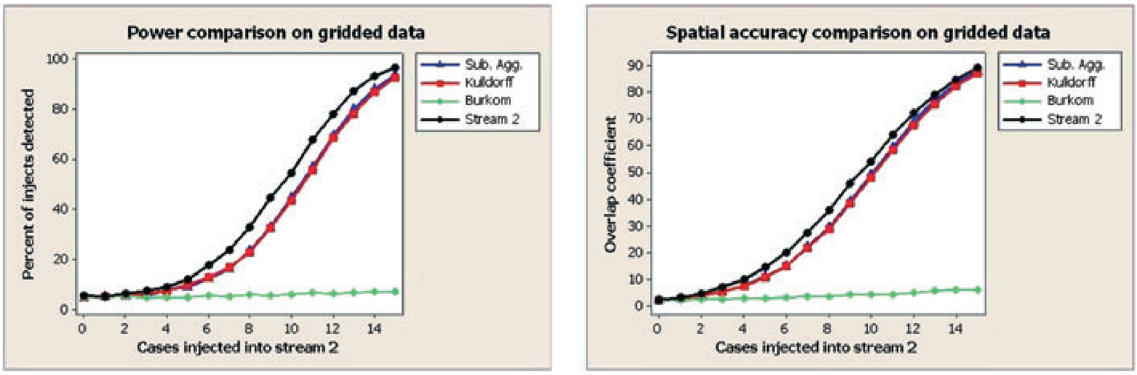

Figure 7. Comparison of detection power and spatial accuracy between Subset Aggregation and Kulldorff multivariate subset scans, assuming $M = 2$ streams of disparate scale, with $x_1 = 0$ for the high-count stream and varying x_2 for the low-count stream. Methods are compared with a simple univariate aggregation approach [4] and a univariate scan monitoring only the affected stream. (a) Percent of injects detected, assuming a 5% false positive rate. (b) Average overlap coefficient between true and detected regions.

the Subset Aggregation and Kulldorff multivariate subset scans were also able to achieve high detection power and accuracy, with no significant differences in performance between the two methods.

7. Comparison of subset aggregation and Kulldorff methods for outbreak detection

We now present an empirical comparison of detection time and spatial accuracy for the Subset Aggregation and Kulldorff multivariate subset scans, using a large set of simulated disease outbreaks injected into 16 streams of real-world ED data from Allegheny County, Pennsylvania. For each method, we compared the fast localized scan approach (searching over proximity-constrained subsets of locations) and the circular scan (searching over the set of overlapping circular regions of varying radius centered at each spatial location). In each case, we used the EBP space–time scan statistic, with a maximum temporal window size of $W_{\max} = 3$. On the basis of the results in §4, we used our fast FF algorithm (with $R = 50$ random restarts) for the Subset Aggregation scan statistic when $k \geqslant 8$ and $M \geqslant 8$. Otherwise, we used our FN algorithm if $k > M$ or our NF algorithm if $k \leqslant M$. On the basis of the results in §5, we used our fast FK algorithm (with $R = 50$ random restarts) for Kulldorff's multivariate scan statistic when $k \geqslant 11$ and the naive NK algorithm otherwise. We now describe the data, outbreak simulations, evaluation metrics, and results in detail.

7.1. *Description of emergency department data*

We obtained a dataset of 612,713 de-identified ED visit records collected from 10 Allegheny County hospitals from January 1, 2004 to December 31, 2005. Each record contains fields for the patient's date of admission to the ED, home zip code, chief complaint (free text), and ICD9 code (numeric). We removed records where the home zip code or admission date was missing or where the home zip code was outside Allegheny County, leaving 397,134 records (64.8%). The free-text chief complaint was present for all remaining records, and the ICD9 code was present for 336,338 (84.7%) of the remaining records. From this dataset, we formed 16 different count data streams d_m ($m = 1 \ldots 16$), each representing a different group of disease symptoms: abdominal pain, bloody stools, botulinic, chest pain, confusion, constitutional, cough, diarrhea, dizziness/fainting, fever, headache, hemorrhaging, nausea/vomiting, rash, runny nose, and sore throat. Each count $c_{i,m}^t$ was calculated by counting the number of cases in zip code s_i on day t that matched the given symptom type d_m, on the basis of either the ICD9 code [29] or the substring matches within the chief complaint text. For example, a patient record was determined to exhibit cough symptoms if its chief complaint string contained the substrings 'cough', 'dyspnea', 'shortness', or 'sob', or if its ICD9 code was equal to 786.2 (cough) or 786.05 (shortness of breath). The set of records was manually refined to remove spurious substring matches. The 16 resulting datasets had mean daily counts ranging from 0.5 to 44.0, with standard deviations ranging from 0.75 to 12.1. The data streams exhibited mild overdispersion, with an average variance-to-mean ratio of 1.70. Variance-to-mean ratios were between 1.1 and 2.2 for 14 of the 16 data streams, 3.13 for fever, and 3.31 for cough. As the disease cases were spread over 97 zip codes, for each day there were many zip codes with no patient visits. Additionally, many streams exhibited both day-of-week and seasonal trends. As noted in [12], a disease surveillance system should be able to reliably detect outbreaks without producing an excessive number of false positive alarms due to the variability in the background data, and thus we believe that our semi-synthetic simulation approach will produce more relevant evaluation results than typical fully synthetic simulations. Although the presence of true disease outbreaks in the real-world dataset could potentially affect our results, no known outbreaks are present in the data. Additionally, as described earlier, we performed additional experiments using a simulated, purely spatial dataset, and similar results were obtained, suggesting that any outbreaks present in the real-world data did not substantially affect our results. Finally, because we conducted the experiments in the following text entirely using the available data from Allegheny County, we acknowledge that results may differ for data from other hospital systems with different concentrations of patient residence zip codes and population densities.

7.2. *Simulation of outbreaks*

We used a semi-synthetic testing framework (injecting simulated disease outbreaks into the real-world ED data) to compare the detection power and spatial accuracy of our methods. We first considered a simple class of simulated outbreaks with a linear increase in the expected number of cases over the duration of the outbreak. More precisely, our outbreak simulator takes three parameters: the outbreak duration

T, the outbreak severity Δ_m for each monitored data stream d_m, and the subset of affected zip codes S_{inject}. Then for each injected outbreak, the outbreak simulator chooses the start date of the outbreak t_{start} uniformly at random. On each day t of the outbreak, $t = 1 \ldots T$, the outbreak simulator increments each count $c^t_{i,m}$ for each affected zip code by Poisson$(t w_{i,m} \Delta_m)$, where $w_{i,m}$ is the 'weight' of that zip code for data stream d_m, $w_{i,m} = \frac{\sum_t c^t_{i,m}}{\sum_i \sum_t c^t_{i,m}}$. We note that the random assignment of inject cases to zip codes is performed separately for each stream, conditioned on the affected region. This can result in a given zip code receiving a non-zero number of injected cases for some data streams and zero cases for other streams.

We considered 10 differently shaped outbreak regions S_{inject}, including approximately equal numbers of circular, elongated, and irregular regions, as shown in Figure 8. For each outbreak region, we created 100 different, randomly generated outbreaks, giving a total of 1000 outbreaks for evaluation. We assumed all outbreaks to be 2 weeks in duration ($T = 14$). The values of Δ_m for each stream d_m were chosen such that the total number of cases for each stream would be increased by one standard deviation on day 7 of the outbreak and two standard deviations on day 14. We performed seven sets of experiments using these synthetic outbreaks: four experiments in which all monitored streams were affected and three experiments in which only a randomly chosen subset of streams were affected; that is, no counts were injected into the unaffected streams. For the first four experiments, we considered $M = 2$, 4, 8, and 16 monitored data streams. For the last three experiments, we considered $M = 8$ data streams, with 1, 2, and 4 affected streams.

We note that simulation of outbreaks is an active area of ongoing research in biosurveillance. The creation of realistic outbreak scenarios is important because of the difficulty of obtaining sufficient labeled data from real outbreaks but is also very challenging. Highly detailed outbreak simulations such as those of Wallstrom *et al.* [30] and Hogan *et al.* [31] combine disease trends observed from past outbreaks with information about the current background data into which the outbreak is being injected as well as allow the user to adjust parameters such as outbreak duration and severity. Although the simple linear outbreak model that we use here is not a realistic model of the temporal progression of an outbreak, it enables precise comparison of the detection power of different methods, gradually increasing the severity of the outbreak until it is detected. As our evaluation did not compare different maximum temporal window sizes (we assumed $W_{\max} = 3$ for all methods), we expect the relative ordering of the methods' detection power to be similar for more realistic outbreak scenarios. However, the absolute differences in detection time will be highly dependent on how quickly the number of outbreak cases 'ramps up' over time: a large difference in power may only result in small differences in detection time if the number of outbreak cases grows quickly, whereas a small difference in power may result in large differences in detection time if the outbreak peaks at a level which is barely detectable by one method and undetectable by the other.

On the basis of these considerations, we rely primarily on the linear outbreak model to compare detection power but present additional evaluation results for one potentially realistic scenario, an increase in respiratory and fever ED cases resulting from an airborne release of anthrax spores. We generated a total of 82 simulated anthrax attacks, using the detailed, realistic simulation model described in [31], and injected these into the Allegheny County ED data. Each simulation generated between 33 and 1324 cases in total (mean = 429.2, median = 430) over a 10-day outbreak period. For these simulations, we assumed all methods to monitor eight data streams including the two affected streams. Thus, our

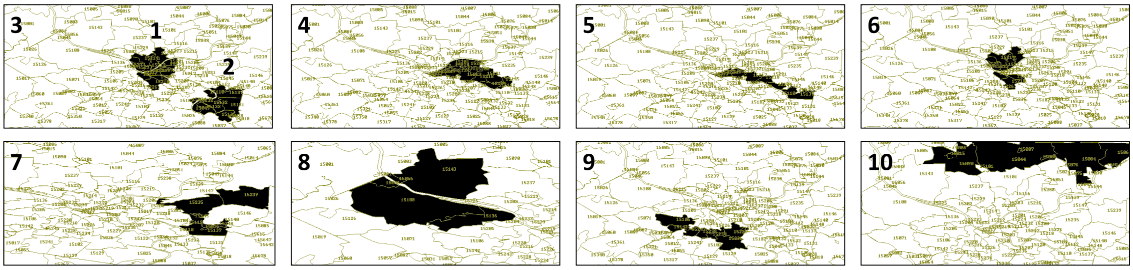

Figure 8. Ten simulated outbreak regions used in our semi-synthetic tests. Note that outbreak region #3 consists of two disjoint, circular clusters; outbreak region #1 is the northwest cluster only, and outbreak region #2 is the southeast cluster only.

scenario represents the case of general, day-to-day public health monitoring as opposed to a targeted system for anthrax attack detection.

7.3. Comparison of detection power

We first computed each method's proportion of outbreaks detected and average number of days to detect, as a function of the allowable false positive rate, for each of the seven experiments with linearly increasing outbreaks. To do this, we computed the maximum region score $F^* = \max_{D,S,W} F(D, S, W)$ for each day of the original dataset with no outbreaks injected. Then for each of the 1000 injected outbreaks, we computed the maximum region score for each outbreak day and determined what proportion of the days for the original dataset have higher scores. Assuming that the original dataset contains no outbreaks, this is the proportion of false positives that we would have to accept in order to have detected the outbreak on day t. For a fixed false positive rate r, the 'days to detect' for a given outbreak is computed as the first outbreak day ($t = 1 \ldots 14$) with proportion of false positives less than r. If no day of the outbreak has proportion of false positives less than r, the method has failed to detect that outbreak: for the purposes of our 'days to detect' calculation, these are counted as 14 days to detect, but could also be penalized further.

Figure 9 compares the timeliness of outbreak detection (average days to detect) between the Subset Aggregation and Kulldorff multivariate subset scans, assuming a fixed false positive rate of one false positive per month. For each method, we considered the fast localized scan with neighborhood sizes k ranging from 5 to 90 as well as the circular scan. As we can see from the figure, both fast localized scan methods achieved significantly faster detection than the two circular scan methods for well-chosen values of k. We note that choosing the optimal neighborhood size is a challenging open problem; however, fast localized scan outperforms circular scan for a wide range of k values and thus can be considered relatively robust to the choice of neighborhood size. In our experiments, detection performance was typically optimized at $k = 10$ or $k = 15$. For the most difficult cases, in which only one or two of the eight monitored data streams were affected, fast localized scan achieved a 1.5-day to 2-day improvement in detection time as compared with circular scan; for the other experiments, detection time was improved between 0.2 and 1.4 days. Comparing the Subset Aggregation and Kulldorff circular scans, we observe that Kulldorff's method was able to detect 0.4 days earlier than Subset Aggregation for $M = 2$ data streams, whereas the two circular scan methods performed similarly for larger values of M. For fast localized scan, Kulldorff's method tended to outperform Subset Aggregation by a small margin: 0.5 days for $M = 2$ and 0.2 to 0.3 days for larger values of M.

These results are also apparent in Figure 10, in which we compare the activity monitoring operating curves (AMOCs) [32] for the Subset Aggregation and Kulldorff methods, using the fast localized scan with $k = 15$ and the circular scan. Each AMOC shows the average detection time for the given method, for false positive rates ranging from 0 to 24 false positives per year. For $M = 2$ data streams, we see a clear ordering of methods, in which Kulldorff's fast localized scan detects fastest, followed by the Subset Aggregation fast localized scan, and the Subset Aggregation circular scan detects the slowest. For larger

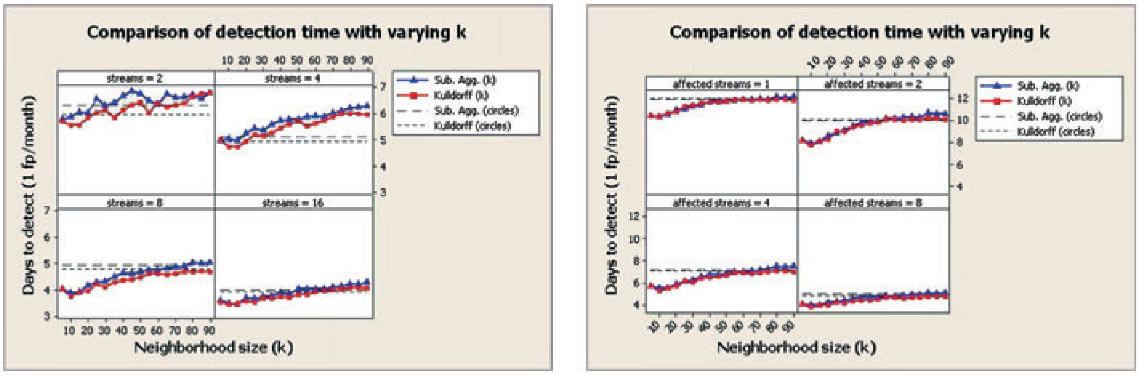

Figure 9. Comparison of detection time between Subset Aggregation and Kulldorff multivariate subset scans, for fast localized scan (as a function of the neighborhood size k) and circular scan. Average number of days to detection at one false positive per month for linearly increasing outbreaks. (a) Results for 2 to 16 data streams, with all streams affected. (b) Results for eight data streams, with one to eight streams affected.

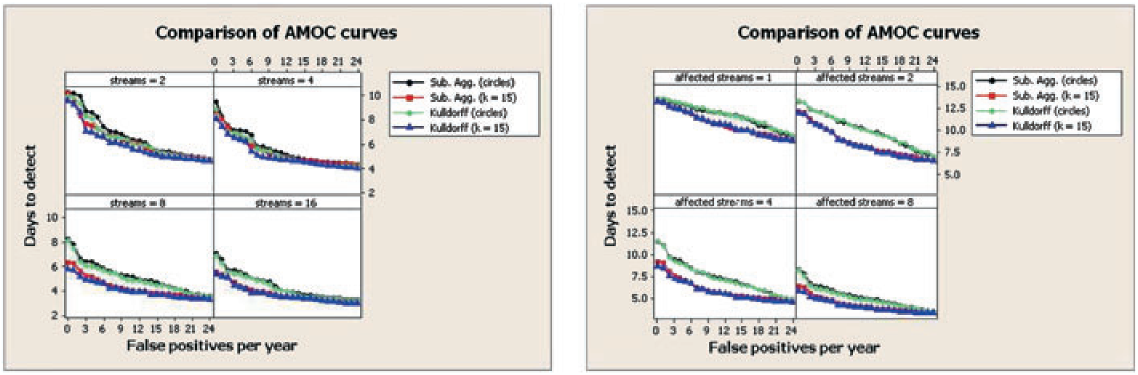

Figure 10. Comparison of activity monitoring operating curves (AMOCs) for Subset Aggregation and Kulldorff multivariate subset scans with fast localized scan ($k = 15$) and circular scan, for linearly increasing outbreaks. (a) Results for 2 to 16 data streams, with all streams affected. (b) Results for eight data streams, with one to eight streams affected.

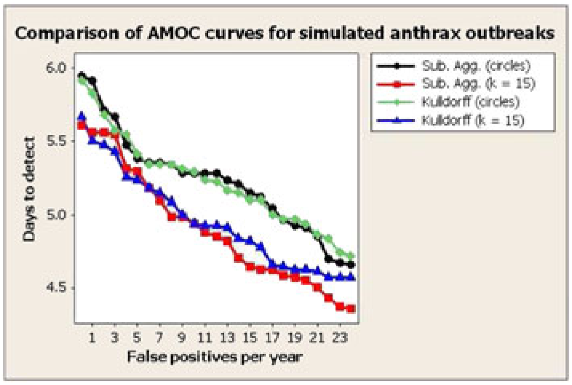

Figure 11. Comparison of activity monitoring operating curves (AMOCs) for Subset Aggregation and Kulldorff multivariate subset scans with fast localized scan ($k = 15$) and circular scan. Results for simulated anthrax attacks (eight monitored data streams, with two streams affected).

values of M and for the cases where only a subset of data streams are affected, the two fast localized scan methods achieve significantly faster detection than the two circular scan methods, but the AMOCs of the Subset Aggregation and Kulldorff methods are similar.

Finally, we compare the AMOC curves for the Subset Aggregation and Kulldorff methods, using the fast localized scan with $k = 15$ and the circular scan, on the 82 simulated anthrax attacks. As shown in Figure 11, the fast localized scan methods achieved consistently and significantly lower time to detect than the corresponding circular scan methods. At a fixed false positive rate of one false positive per month, we observe improvements in detection time of 0.4 and 0.3 days for Subset Aggregation and Kulldorff's method, respectively. We note that the absolute improvements in detection time were smaller for the simulated anthrax attacks as compared with the linearly increasing outbreaks because the number of injected cases ramped up more quickly and all methods had lower average time to detect. We observed no significant differences in detection time between Subset Aggregation and Kulldorff's method. Thus, our experiments demonstrate two main results: small improvements in detection power for Kulldorff's method as compared with Subset Aggregation in some outbreak scenarios, and consistently large improvements in detection power for fast localized scans as compared with the traditional circular scan approach.

7.4. *Comparison of spatial accuracy*

In addition to the comparison of detection times described earlier, we also computed the average spatial accuracy (degree of overlap between true and detected clusters) for each method for each outbreak day. We averaged results over all outbreaks for each of the experiments discussed earlier.

Letting S^* represent the detected region $S^* = \arg\max_S F(D, S, W)$ and S_T represent the true inject region (the subset of locations for which simulated cases were actually injected), we define the *spatial overlap coefficient* as: Overlap $= \frac{\sum_{s_i \in S^* \cap S_T} w_i}{\sum_{s_i \in S^* \cup S_T} w_i}$, where w_i is the weight of location s_i. Each weight w_i was set proportional to the total number of disease cases observed in that location, which can also be thought of as a proxy for the at-risk population; thus, our measure emphasizes a method's ability to detect locations with a larger case count (e.g., more densely populated zip codes). The overlap coefficient can vary between 0 and 1, with Overlap $= 1$ if $S^* = S_T$ and Overlap $= 0$ if S^* and S_T are disjoint.

For the seven experiments with linearly increasing outbreaks, Figure 12 compares the spatial overlap coefficients for the Subset Aggregation and Kulldorff multivariate subset scans at the midpoint of the outbreak. For each method, we considered the fast localized scan with neighborhood sizes k ranging from 5 to 90, as well as the circular scan. As we can see from the figure, both fast localized scan methods achieved significantly higher spatial accuracy than the two circular scan methods for well-chosen values of k. For $M = 16$ data streams, fast localized scans with $k = 15$ achieved overlap coefficients of 83%, as compared with 70% for the circular scans. For smaller numbers of affected data streams and for suboptimal values of k, the improvements were smaller, and in the most difficult case (where only one of the eight monitored data streams was affected), the overlap coefficients of the fast localized scans were 2% lower than the circular scans. There were no significant differences in spatial accuracy between the Subset Aggregation and Kulldorff circular scans. Similarly, for fast localized scan, we observed only small differences in performance: Kulldorff's method achieved 1–1.5% higher accuracy than Subset Aggregation for low values of k and small numbers of affected data streams, whereas Subset Aggregation achieved 0.5–1% higher accuracy than Kulldorff's method for high values of k and large numbers of affected data streams. Figure 13 compares the spatial overlap coefficients of the Subset Aggregation and Kulldorff methods over the entire duration of the outbreak for fast localized scans with $k = 15$ and circular scans. As we can see from the figure, all methods performed similarly when one or two data streams were affected, whereas for four or more affected data streams, the two fast localized scans outperformed the two circular scans starting from the fourth outbreak day.

Finally, Figure 14 compares the spatial accuracy of the Subset Aggregation and Kulldorff methods for the 82 simulated anthrax attacks. For each method, we considered the fast localized scan with neighborhood sizes k ranging from 5 to 90, as well as the circular scan, and computed the spatial overlap coefficient at the midpoint of the outbreak. We observe that, for both Subset Aggregation and Kulldorff's method, the fast localized scan achieved significantly higher accuracy than the circular scan for all neighborhood sizes $k > 15$. Kulldorff's method achieved slightly higher accuracy than Subset Aggregation, but for most values of k these differences were not significant. These results suggest that, while there are no major differences in spatial accuracy between the Subset Aggregation and Kulldorff multivariate subset scans, both methods can achieve substantially increased accuracy by searching over proximity-constrained subsets rather than circles, using our fast multivariate scan approaches.

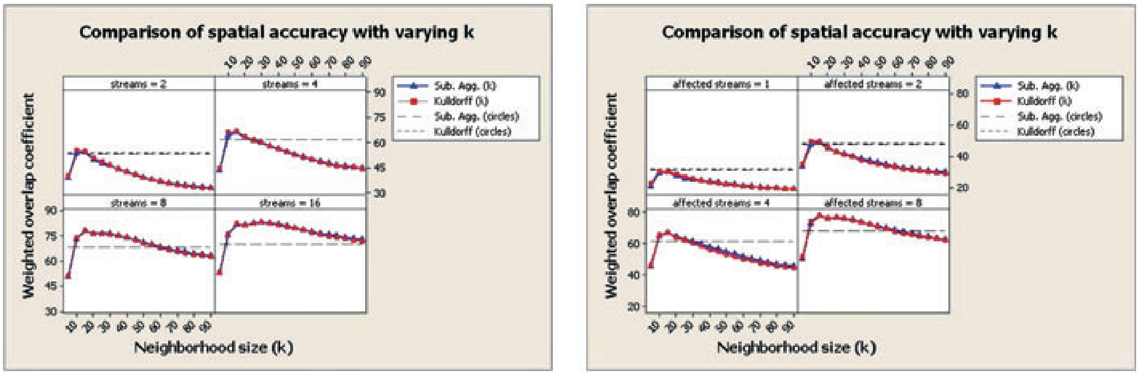

Figure 12. Comparison of spatial accuracy between Subset Aggregation and Kulldorff multivariate subset scans, for fast localized scan and circular scan. Weighted overlap coefficient between true and detected spatial regions, at the midpoint of the outbreak, for linearly increasing outbreaks. (a) Results for 2 to 16 data streams, with all streams affected. (b) Results for eight data streams, with one to eight streams affected.

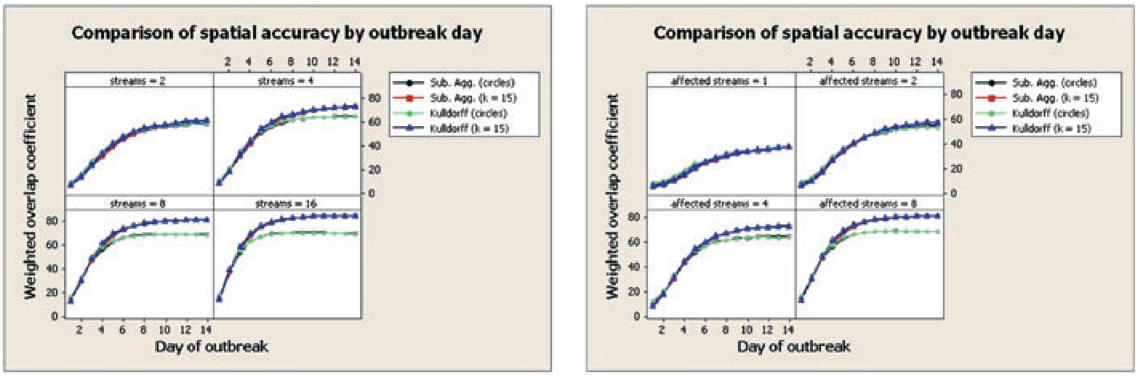

Figure 13. Comparison of spatial accuracy as a function of the outbreak day, for Subset Aggregation and Kulldorff multivariate subset scans with fast localized scan ($k = 15$) and circular scan. Weighted overlap coefficient between true and detected spatial regions, for linearly increasing outbreaks. (a) Results for 2 to 16 data streams, with all streams affected. (b) Results for eight data streams, with one to eight streams affected.

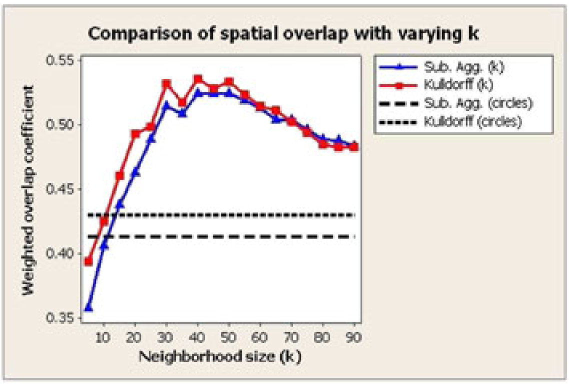

Figure 14. Comparison of spatial accuracy between Subset Aggregation and Kulldorff multivariate subset scans, for fast localized scan and circular scan. Weighted overlap coefficient between true and detected spatial regions, at the midpoint of the outbreak. Results for simulated anthrax attacks (eight data streams, with two streams affected).

7.5. *Comparison of stream detection accuracy*

For each of the four experiments with eight monitored data streams and varying numbers of affected streams, we evaluated how well each method was able to correctly identify the affected subset of streams. We computed the average overlap coefficient between the true and detected sets of streams for each method for each outbreak day ($t = 1 \ldots 14$). Letting D_T represent the true set of affected data streams and D^* represent the set of detected streams, $D^* = \arg\max_D F(D, S, W)$, we define the *stream overlap coefficient* as Overlap $= \frac{|D_T \cap D^*|}{|D_T \cup D^*|}$. The overlap coefficient can vary between 0 and 1, with Overlap $= 1$ if $D_T = D^*$ and Overlap $= 0$ if D_T and D^* are disjoint.

Figure 15 compares the stream overlap coefficients for the Subset Aggregation and Kulldorff multivariate subset scans at the midpoint of the outbreak. As we can see from the figure, Subset Aggregation achieved significantly (11–17%) higher stream detection accuracy than Kulldorff's method for the cases of one, two, and four affected data streams; because Kulldorff's method tends to report more data streams than Subset Aggregation does, it had slightly higher accuracy for the trivial case where all eight data streams were affected. The neighborhood size k did not substantially affect stream detection accuracy for either method, although accuracy was slightly decreased for very small values of k. In Figure 16, we considered fast localized scans with a neighborhood size of $k = 15$, comparing the stream overlap coefficients of the Subset Aggregation and Kulldorff methods over the entire duration of the outbreak. When only a subset of streams were affected, Subset Aggregation achieved significant improvements in stream detection accuracy starting from the fourth or fifth outbreak day. Similarly, Subset Aggregation achieved

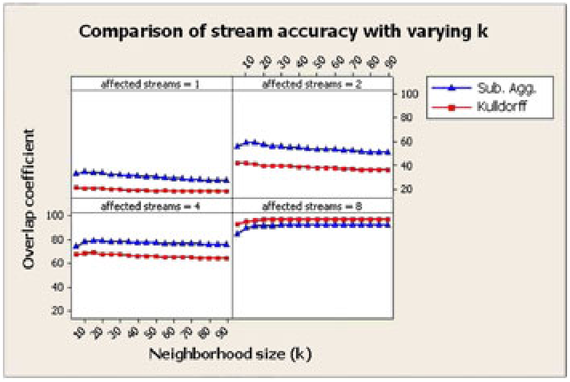

Figure 15. Comparison of stream accuracy between Subset Aggregation and Kulldorff multivariate subset scans, using fast localized scan, as a function of the neighborhood size k and the number of affected streams. Overlap coefficient between true and detected streams, at the midpoint of the outbreak.

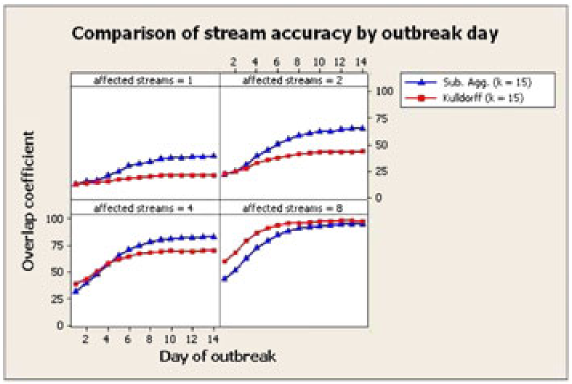

Figure 16. Comparison of stream accuracy between Subset Aggregation and Kulldorff multivariate subset scans, using fast localized scan with neighborhood size $k = 15$, as a function of the outbreak day and the number of affected streams. Overlap coefficient between true and detected streams.

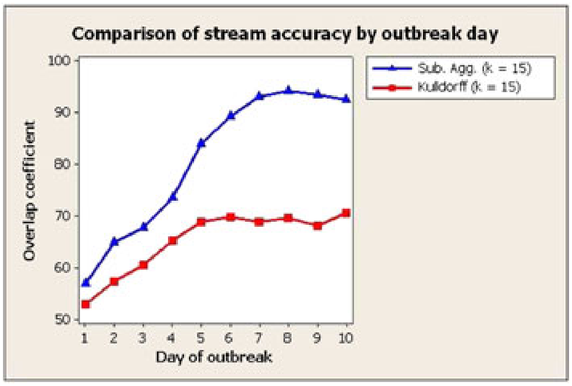

Figure 17. Comparison of stream accuracy between Subset Aggregation and Kulldorff multivariate subset scans, using fast localized scan with neighborhood size $k = 15$, as a function of the outbreak day. Overlap coefficient between true and detected streams on 82 simulated anthrax attacks.

significant improvements in stream detection accuracy as compared with Kulldorff's method for the 82 simulated anthrax attacks, as shown in Figure 17. The primary reason for these differences in accuracy, as discussed earlier, was that Kulldorff's method tended to report both affected and unaffected data streams, whereas Subset Aggregation was better able to discriminate affected from unaffected streams.

8. Conclusions

This paper has presented several contributions to the literature on multivariate event detection. First, we proposed efficient algorithms to identify the highest-scoring subsets of locations and data streams in multivariate space–time data, thus making multivariate spatial scan statistics computationally feasible for detecting irregularly shaped space–time clusters in massive, high-dimensional datasets. These algorithms extend our previously proposed 'fast subset scan' framework [12] from univariate to multivariate space–time data, exploiting the 'LTSS' property of many commonly used scan statistics in order to optimize over the exponentially many subsets of the data while only evaluating a linear number of subsets. In this work, we considered two variants of the multivariate subset scan: Subset Aggregation, which assumes a constant risk across the subset of affected data streams, and Kulldorff's multivariate scan [5], which assumes independent risks for each stream. Subset Aggregation aggregates counts and baselines across subsets of streams and then applies the univariate log-likelihood ratio statistic to these aggregates, whereas Kulldorff's method computes a separate log-likelihood ratio statistic for each data stream and then adds scores across streams. As a result, we need different algorithms to make each method computationally efficient and scalable. In this work, we developed fast algorithms for both Subset Aggregation and Kulldorff methods, making both approaches computationally feasible for optimization over many locations and many data streams. For Subset Aggregation, our FF algorithm iterates between optimizing over subsets of locations and subsets of streams, whereas for Kulldorff's method, our FK algorithm iterates between optimizing over subsets of locations and estimating the underlying risk for each data stream. We also proposed two additional fast algorithms for Subset Aggregation, FN (for datasets with a small number of streams) and NF (for datasets with a small number of locations), and we performed an empirical comparison of fast and naive algorithms for both Subset Aggregation and Kulldorff methods. As in the univariate case, we can easily incorporate spatial proximity constraints into our fast subset scan framework, efficiently maximizing the log-likelihood ratio statistic over spatial clusters of nearby locations as well as subsets of data streams.

We used our fast search algorithms to enable a detailed comparison of the Subset Aggregation and Kulldorff multivariate subset scans for synthetic and real-world disease surveillance datasets. For both methods, we compared our 'fast localized scan' approach (searching over all subsets of locations constrained by spatial proximity) to the traditional spatial scan approach (searching over circular regions). We demonstrated that the fast localized scan significantly improved detection power and spatial accuracy for both Subset Aggregation and Kulldorff methods while maintaining efficient and scalable computation. These empirical results show the efficacy of our fast multivariate subset scan approach (scanning over proximity-constrained subsets) as compared with the traditional, circular scan.

Comparing the Subset Aggregation and Kulldorff methods, we demonstrated that Kulldorff's method tends to achieve somewhat higher detection power and spatial accuracy when data streams are affected to differing extents. However, when only a subset of streams are affected, Subset Aggregation more accurately characterizes events by identifying the affected subset of streams. Thus, Kulldorff's method may be preferable when event detection is the primary goal, whereas Subset Aggregation may be preferable when event characterization is paramount. Our fast, scalable algorithms enable either method to be effectively applied to massive, high-dimensional datasets.

Our future work will extend the fast multivariate subset scan framework in several directions. First, LTSS can be used to accelerate spatial scans with other constraints, including shape and connectivity. In each case, we have integrated LTSS into a 'branch and bound' framework, using the unconstrained 'all subsets' score of a group of locations as an upper bound on the constrained score and ruling out many subsets of locations that are probably suboptimal. Our recently proposed GraphScan algorithm [33] enables efficient maximization of scan statistics over all connected subsets for graphs of over 100 nodes, and the techniques described here can be used to extend GraphScan from univariate to multivariate datasets. Second, our FF algorithm for the Subset Aggregation scan statistic can be thought of as an optimization over subsets of a matrix (second-order tensor), where the two tensor modes represent spatial locations and data streams, respectively, and we wish to identify an optimal subset for each mode. We can generalize FF to higher-order tensors as well: for example, given the observed and expected counts of disease cases for each symptom type d_m in each spatial location s_i for each of multiple demographic groups, we could simultaneously optimize over subsets of symptoms, proximity-constrained subsets of locations, and subsets of demographic groups, thus more precisely identifying the affected subpopulations. Finally, we have extended LTSS to non-spatial datasets, where we have a set of attributes for each data record, and wish to detect self-similar groups of records that have anomalous values for some subset of attributes. We have recently proposed the 'fast generalized subset scan' algorithm [34]. Fast generalized subset scan learns the structure and parameters of a Bayesian network from the data, computes empirical p-values corresponding to each attribute value, and efficiently maximizes a nonparametric scan statistic [6] over all subsets of records and attributes, subject to the constraints on group self-similarity.

Acknowledgement

We gratefully acknowledge funding support from the National Science Foundation grants IIS-0916345, IIS-0911032, and IIS-0953330.

References

1. Kulldorff M. A spatial scan statistic. *Communications in Statistics: Theory and Methods* 1997; **26**(6):1481–1496.
2. Wagner MM, Moore AW, Aryel R (eds). *Handbook of Biosurveillance*. Elsevier: New York, NY, 2006.
3. Burkom HS, Murphy SP, Coberly J, Hurt-Mullen K. Public health monitoring tools for multiple data streams. *Morbidity and Mortality Weekly Report* 2005; **54**((Supplement)):55–62.
4. Burkom HS. Biosurveillance applying scan statistics with multiple, disparate data sources. *Journal of Urban Health* 2003; **80**(2 Suppl. 1):i57–i65.
5. Kulldorff M, Mostashari F, Duczmal L, Yih WK, Kleinman K, Platt R. Multivariate scan statistics for disease surveillance. *Statistics in Medicine* 2007; **26**:1824–1833.
6. Neill DB, Lingwall J. A nonparametric scan statistic for multivariate disease surveillance. *Advances in Disease Surveillance* 2007; **4**:106.
7. Neill DB, Cooper GF. A multivariate Bayesian scan statistic for early event detection and characterization. *Machine Learning* 2010; **79**:261–282.
8. Patil GP, Taillie C. Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environmental and Ecological Statistics* 2004; **11**:183–197.
9. Duczmal L, Assuncao R. A simulated annealing strategy for the detection of arbitrary shaped spatial clusters. *Computational Statistics and Data Analysis* 2004; **45**:269–286.
10. Rolka H, Burkom H, Cooper GF, Kulldorff M, Madigan D, Wong W-K. Issues in applied statistics for public health bioterrorism surveillance using multiple data streams: research needs. *Statistics in Medicine* 2007; **26**:1834–1856.
11. Freifeld CC, Mandl KD, Reis BY, Brownstein JS. HealthMap: global infectious disease monitoring through automated classification and visualization of Internet media reports. *Journal of the American Medical Informatics Association* 2008; **15**:150–157.
12. Neill DB. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society (Series B: Statistical Methodology)* 2012; **74**(2):337–360.
13. Neill DB, Moore AW, Sabhnani MR, Daniel K. Detection of emerging space–time clusters. *Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, 2005.
14. Kulldorff M, Nagarwalla N. Spatial disease clusters: detection and inference. *Statistics in Medicine* 1995; **14**:799–810.
15. Kulldorff M, Feuer EJ, Miller BA, Freedman LS. Breast cancer clusters in the northeast United States: a geographic analysis. *American Journal of Epidemiology* 1997; **146**(2):161–170.
16. Hjalmars U, Kulldorff M, Gustafsson G, Nagarwalla N. Childhood leukemia in Sweden: using GIS and a spatial scan statistic for cluster detection. *Statistics in Medicine* 1996; **15**:707–715.
17. Mostashari F, Kulldorff M, Hartman JJ, Miller JR, Kulasekera V. Dead bird clustering: a potential early warning system for West Nile virus activity. *Emerging Infectious Diseases* 2003; **9**:641–646.
18. Neill DB. Detection of spatial and spatio-temporal clusters. *Technical Report CMU-CS-06-142*, Ph.D. thesis, Carnegie Mellon University, School of Computer Science, 2006.
19. Neill DB, Moore AW. Rapid detection of significant spatial clusters. *Proceedings of 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Seattle, WA, 2004; 256–265.
20. Kulldorff M, Huang L, Pickle L, Duczmal L. An elliptic spatial scan statistic. *Statistics in Medicine* 2006; **25**:3929–3943.
21. Tango T, Takahashi K. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics* 2005; **4**:11.
22. Kulldorff M, Athas W, Feuer E, Miller B, Key C. Evaluating cluster alarms: a space–time scan statistic and cluster alarms in Los Alamos. *American Journal of Public Health* 1998; **88**:1377–1380.
23. Kulldorff M. Prospective time-periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society A* 2001; **164**:61–72.
24. Neill DB. An empirical comparison of spatial scan statistics for outbreak detection. *International Journal of Health Geographics* 2009; **8**:20.
25. Neill DB, Moore AW, Cooper GF. A Bayesian spatial scan statistic. *Advances in Neural Information Processing Systems* 2006; **18**:1003–1010.
26. Neill DB. Fast Bayesian scan statistics for multivariate event detection and visualization. *Statistics in Medicine* 2011; **30**(5):455–469.
27. Duczmal L, Cancado A, Takahashi R, Bessegato L. A genetic algorithmic for irregularly shaped scan statistics. *Computational Statistics and Data Analysis* 2007; **52**(1):43–52.
28. Walther G. Optimal and fast detection of spatial clusters with scan statistics. *Annals of Statistics* 2010; **38**(2):1010–1033.
29. World Health Organization. International Classification of Diseases, Ninth Revision (ICD-9).

30. Wallstrom GL, Wagner MM, Hogan WR. High-fidelity injection detectability experiments: a tool for evaluation of syndromic surveillance systems. *Morbidity and Mortality Weekly Report* 2005; **54**((Supplement)):85–91.
31. Hogan WR, Cooper GF, Wallstrom GL, Wagner MM, Depinay JM. The Bayesian aerosol release detector: an algorithm for detecting and characterizing outbreaks caused by atmospheric release of *Bacillus anthracis*. *Statistics in Medicine* 2007; **26**:5225–5252.
32. Fawcett T, Provost F. Activity monitoring: noticing interesting changes in behavior. *Proceedings of 5th International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999; 53–62.
33. Speakman S, Neill DB. Fast graph scan for scalable detection of arbitrary connected clusters. *Proceedings of 2009 International Society for Disease Surveillance Annual Conference*, Miami, FL, 2010.
34. McFowland III E, Speakman S, Neill DB. Fast generalized subset scan for anomalous pattern detection. *Proceedings of INFORMS Annual Conference*, Austin, TX, 2010.