

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Spatio-Temporal Disease Surveillance

Ross Sparks, Sarah Bolt and Chris Okugami  
*CSIRO Mathematics, Informatics and Statistics, Sydney,  
Australia*

## 1. Introduction

Concern over bio-terrorism has led to a demand for automated methods for the surveillance of disease counts with the ability to rapidly detect outbreaks of disease. While traditional statistical process control methods such as control charts have been found to have early detection properties when monitoring univariate disease counts, these are often inadequate for detecting bio-terrorism events.

There are two principal impediments in statistical process control methods for the detection of bio-terrorism events: firstly, these methods aggregate over space by examining total counts and thus ignore the spatial dimension of the task and secondly they fail to adjust for the usual (seasonal) behaviour of diseases (e.g., Steiner *et al.*, 2011 where the focus is early detection of the start of influenza outbreak). If disease outbreaks were expected to be spread relatively uniformly in space, then the former reason is unimportant. However, since bio-terrorism attacks are likely to be introduced to specifically targeted locations, then the resulting disease instances are likely to cluster in space. Consequently, monitoring total counts is likely to reduce the signal-to-noise ratio of this outbreak by aggregating over regions where there is no outbreak. Exploiting the spatial clustering should be able to provide additional power and efficiency in detecting the outbreak.

There is much ongoing work in developing methods that are efficient at detecting outbreaks in a spatial context but these methods may still fail to deal with the latter fault mentioned above. When surveillance for bio-terrorism involves the monitoring of diseases or syndromes already present in the population then the detection of an attack may be delayed if it is introduced during a period of normal seasonal increased activity. For example, respiratory complaints are often much more frequent in the cooler months so detecting an intentional disease outbreak with respiratory symptomatology would need to differentiate between the usual and an unusual increase in cases. Therefore, it is often necessary to remove the influence of the expected behaviour of a disease to detect the signal of an introduced strain early.

Before presenting the method we are proposing in this chapter to deal with both of the above concerns, we begin by outlining some of the existing spatial disease detection methods. The current benchmark in spatio-temporal surveillance is the spatial SCAN statistic (Tango, 1995, Kulldorff, 1995, 1997, 2001, 2005). This method is a spatio-temporal moving average plan that systematically scans the target space applying a test to all windows of data up to a given fixed size in time and space. This presents an intuitive

approach but has received some criticisms by Woodall *et al.* (2008) and Han *et al.* (2008), including: firstly that it is not as efficient as the cumulative SUM (CUSUM) (see Raubertas, 1989, and Rogerson and Yamada, 2004) for outbreak detection and secondly that its ability to detect outbreaks most effectively is dependent on the choice of shape and size of the scanning window. In addition, in some of the literature on the SCAN statistic, the focus is on whether diseases cluster rather than on early detection.

Some control chart methods have also been proposed such as the Multivariate Exponentially Weighted Moving Average (MEWMA) control chart method proposed by Joner *et al.* (2008). This approach could be extended to the lattice structure of counts discussed below but involving all cell counts in the lattice making it a high dimensional application of the MEWMA plan which is difficult to implement. Exponential weighted moving averages (EWMA) have also proved valuable in the early detection of persistent disease outbreaks (Steiner *et al.*, 2010 and Sparks *et al.* 2010a). However these methods are frequently unable to account for underlying changes that occur in the observation of disease instances through health services such as day-of-the-week or holiday effects. The removal of known trends, called nuisance variables, would reduce the noise in the departures of counts from expected, and as such provides the surveillance plans with a increased chance of early detection.

A bio-terrorist attack is likely to be initiated at several simultaneous disconnected locations causing clustered outbreaks of different size and shape. Having a technology that has the flexibility of finding outbreaks started at several disconnected locations is essential in this setting.

In this chapter, we explore the method EWMA Surveillance Trees that addresses the concerns outlined above and that has this flexibility. The approach taken here is to divide the geographical regions into a lattice structure so that the expected numbers of cases within the cells are as similar as possible. Expected counts are achieved by firstly building a model that describes the usual behaviour of disease counts (including any usual seasonal increases) and then using this model to produce a one day-ahead forecast for the expected counts (see Sparks, *et al.* 2010a). The method proposed then:

- Builds temporal memory of the disease process by smoothing the time series of cell counts using MEWMAs. This allows the method to build up memory of the process to detect slowly evolving changes.
- Uses exponentially weighted spatially smoothing of these temporal EWMA cell counts by smoothing across both the rows and columns of the lattice (very similar to MEWMA).
- Detects departures from expected value in the spatio-temporally smoothed lattice counts using a binary recursive partitioning approach that focuses in on specific areas of concern.
- Prunes these partitions in a way that leaves only the outbreaks, that is areas of significant departure from expected.

So this method should be able to detect outbreaks of varying size and shapes and incorporate the benefits of EWMA smoothing for early detection. Its performance will be measured against an implementation of the SCAN statistic.

The EWMA Surveillance Tree method proposed here also applies a smoothing in the spatial sense. If smoothing has the advantage of reducing spatial noise and more easily exposes spatial trends in disease outbreaks, then this could translate into earlier detection properties. This chapter will also evaluate when it is beneficial to spatially smooth counts for the early detection of diseases.

We will continue by giving a detailed overview of the spatial SCAN statistic. We will then propose the EWMA Surveillance Tree methodology in more detail and discuss the results of some simulation studies comparing this method to the SCAN statistic. Lastly we will briefly discuss the extension of these methods to a broader multivariate context which is the key in applications of these methods to syndromic surveillance for emerging diseases.

## 2. The SCAN statistic

Scan statistics are so described because they arise from the scanning of time and/or space looking for clusters of events. This section will present the details for an implementation of the spatial SCAN statistic that will be used as the benchmark in the simulation studies to follow. The SCAN statistic is similar to a spatio-temporal moving average plan that looks at the number of observations in cylinders of spatio-temporal space (e.g., see Figure 1), where the height of the cylinder is time. The SCAN plan used in this chapter is indistinguishable from the approaches used in Glaz *et al.* (2001) in the spatial dimension but uses a moving window for the temporal dimension. As such the results from Chen and Glaz can be used to get efficient starting estimates for the threshold used in the surveillance plan. For other literature on the SCAN plan see Tango (1995), Kulldorff and Nagarwalla (1995), Kulldorff (1997), and Kulldorff, M. (2001), Kulldorff *et al.* (2005), Woodall *et al.* (2008) and Han *et al.* (2008).

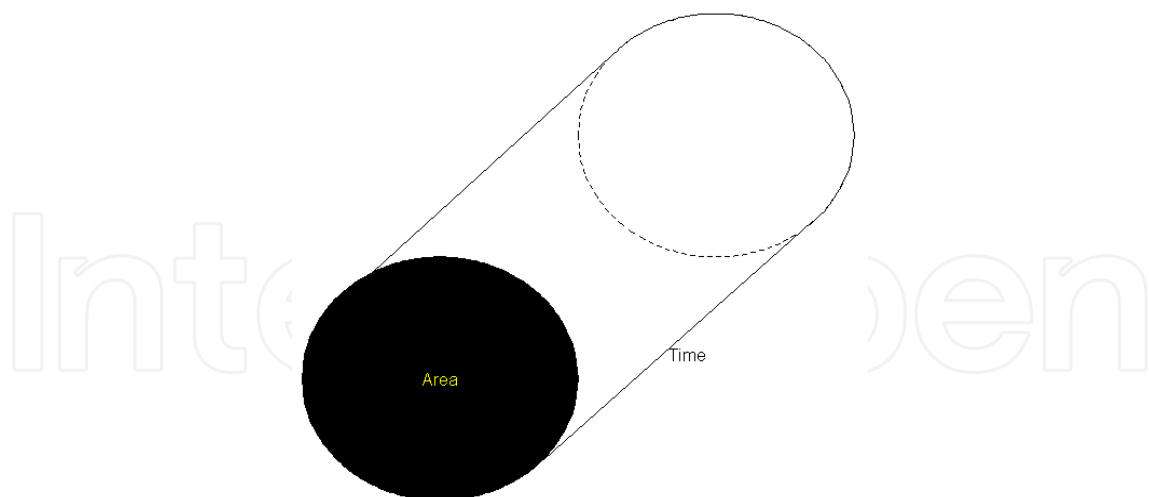


Fig. 1. Example of the scanned region – circular area is scanned and the depth is the number of days that counts are aggregated over.

The SCAN plan in this chapter is applied to a fixed lattice structure over the target area of the data for reasons which will become obvious later. Although this lattice structure may not be regular as in Figure 2 below, it is convenient at this stage to make it regular. In other words, in this lattice, individual cells are not taken to have a fixed volume; they are

constructed so that the marginal total across each row/column spatial dimension has approximately the same expected number of counts. In addition the horizontal and vertical dividers of cells may be curved giving the design greater flexibility.

The lattice structure reduces the amount of multiple testing by forcing the scanned area to always include complete spatial ("square") cells. However, if a bio-terrorism event spans parts of some cells then some information about its signal is lost. The time window is arbitrarily taken as 10 days in this chapter, but could be taken as 7 days or 14 days to do away with the need for accounting for the influence of day-of-the-week effects.

The SCAN plan counts the number of disease cases within the scanned space-time window and compares this count to its expected count, assuming a Poisson distribution. Often autocorrelation of counts is ignored in this process which does cause inefficiencies in the inference of detecting bio-terrorism events. In this chapter we will also ignore the autocorrelation in the temporal counts. Since the SCAN plan involves significant amounts of multiple testing, we need to establish the false alarm rates for the level of multiple testing.

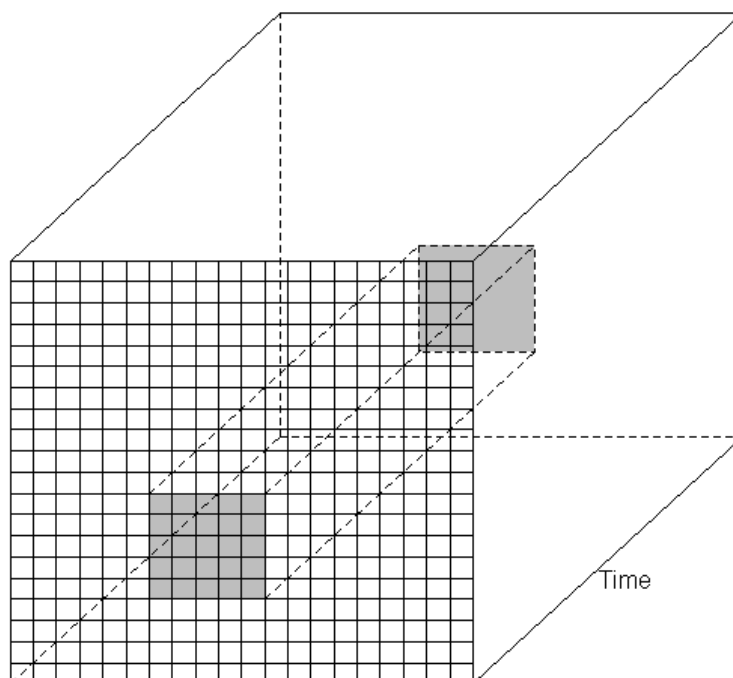


Fig. 2. The target region for assessing outbreaks is 20 by 20 cells by time and the scanned area is all 5 by 5 cells by time volumes - one example of which is illustrated in grey.

In this paper, we use a 40 by 40 cell target region. We examined two levels of spatial memory in the SCAN plans applied later in the chapter. The spatial cells scanned are either 5 by 5 or 10 by 10 both with a moving time window of 10 days. The 5 by 5 cell scan involves  $36 \times 36 = 1296$  different (but overlapping) scanned regions, and the 10 by 10 cell scan involves 961 different (but overlapping) scanned regions. Clearly, the 10 by 10 cell scan involves fewer tests and so its level of significance for the fixed false alarm rate is larger than the 5 by 5 cell scan. This means that it should detect an outbreak close to 10 by 10 in dimensions earlier than the 5 by 5 cell scan. Therefore if the dimension of a future

bio-terrorism outbreak is known in advance then it is best to use these dimensions precisely in designing the scanning window, unless the outbreak is sparse within the region. In this case a smaller window is needed (see Sparks, 2011c) to exploit this sparsity. Unfortunately, we seldom know the dimension of bio-terrorism outbreaks, and therefore we need methodology that is flexible enough for detecting unknown outbreaks quickly. The next section discusses such flexible technology that is not as dependent on the shape and size of the outbreak.

### 3. EWMA surveillance trees

The EWMA Surveillance Tree method proposed here for comparison against the SCAN plan uses an EWMA approach by smoothing counts first temporally to build in temporal memory of the process, and then spatially to average counts locally in space. Smoothing can also be viewed as an approach that reduces the noise in counts without reducing the signal too much. This makes the signal-to-noise ratio larger and therefore has early detection advantages. There are many examples of EWMA temporal smoothing of counts data aimed at early detection of diseases, for example see Sparks *et al.* (2010a) and Stienen *et al.* (2010) for Poisson counts, and Sparks *et al.* (2010b) and Sarka (2011) for negative binomial counts. There is no literature on spatial exponentially weighted moving average smoothing although there is some work by Rogerson *et al.* (2006) using 2 dimensional kernel smoothers. The smoothed counts are then used in an outbreak detection procedure.

The EWMA smoothing process has computational advantages because only the last temporally smoothed cell counts are retained rather than the last 10 days as is necessary in the SCAN plan discussed in the previous section. Areas are searched for outbreaks by comparing their smoothed observed counts to their respective smoothed expected counts. An outbreak is signalled when the difference between observed and expected counts in a window exceeds a threshold. The value of the threshold is chosen to control the false alarm rate. Note that EWMA smoothed counts are no longer Poisson distributed, however EWMA smoothing (spatial and temporal) helps improve the approximation to a normal distribution of the smoothed counts.

The method consists of three major steps:

1. EWMA based smoothing of observed and expected counts, first temporally then spatially;
2. Growing a surveillance tree of departures from expected value in the spatio-temporally smoothed counts using a binary recursive partitioning approach
3. Pruning the surveillance tree to reveal outbreaks and control the false alarm rate.

Each of these steps is detailed in subsections 3.1, 3.2 and 3.3 respectively.

#### 3.1 EWMA temporal and spatial smoothing

Define 40 by 40 matrices of counts and expected values for day  $t$  by  $Y_t = \{y_{(\ell,k,t)}\}$  and  $\Lambda_t = \{\lambda_{(\ell,k,t)}\}$  respectively. Temporal smoothing examines the exponentially weighted moving average (EWMA) to the counts according to

$$\bar{Y}_t = \alpha Y_t + (1 - \alpha) \bar{Y}_{t-1}$$

( $\bar{Y}_0 = \Lambda_1$ ) and similarly smooths the matrix of expected values of these counts using

$$\bar{\Lambda}_t = \alpha \Lambda_t + (1 - \alpha) \bar{\Lambda}_{t-1}$$

where  $\bar{\Lambda}_0 = \Lambda_1$ , and  $0 < \alpha < 1$  is a constant that determines how much memory to retain in the averages. Larger values of  $\alpha$  retain less temporal memory.

Once the counts have been smoothed temporally, we then smooth over space to average counts locally. So the temporal smoothed counts and temporal smoothed expected values of  $\bar{Y}_t$  and  $\bar{\Lambda}_t$  respectively are now spatially smoothed using smoothing matrix defined by  $\{A_{ij}\} = \{(1 - \lambda)^{|i-j|} / \sum_{j=1}^{40} (1 - \lambda)^{|i-j|}\}$ . The spatial smoothing of counts is defined by

$$\tilde{Y}_t = A \bar{Y}_t A^t$$

and the spatial smoothing of their expected values is defined by

$$\tilde{\Lambda}_t = A \bar{\Lambda}_t A^t$$

This EWMA spatial smoothing has the advantage over other smoothers of computational simplicity and is efficient in terms of early outbreak detection. Alternative spatial smoothing involving two dimensional kernel smoothers (Riedel, 1993, Rogerson *et al.* 2006) or splines (Currie *et al.* 2003, Lee and Durban, 2009) are more complex computationally. The EWMA smoother is similar to a kernel smoother using a double exponential distribution. The main reason for this choice is computational speed. We ignore the boundary problems with this EWMA smoothing approach. It is clear that the spatial smoothing employed here smooths boundary cell counts less. In addition, the recursive partitioning approach does not adjust for the differential smoothing. Therefore the surveillance trees detect outbreaks that hug a boundary very quickly. For this reason no clusters are considered that hug boundaries in the simulation section, because this would exaggerate the performance of surveillance trees.

### 3.2 Growing the surveillance trees

The recursive partitioning approach applied in this paper is very similar to that in Yu (2009) and identical to Sparks and Okugami (2010b). Surveillance trees generate offspring by recursively partitioning either longitudinal cells or latitudinal cells into rectangular regions. We start by finding the best partition of longitudinal cells in terms of having the highest signal to noise ratio, and then the best partition of latitudinal cells. Thereafter, select the partition with the highest signal-to-noise ratio between the best longitudinal cell partition and the best latitudinal cell partition.

The choice of the signal-to-noise ratio as the measure of departure from expected value was decided based on the fact that the square root of the smooth counts has roughly a constant variance and its mean is equal to the square root of the smoothed mean.

The mathematical summary of this process for the first generation of offspring is as follows:

denote  $\tilde{Y}_i = \sum_{j \in P} \tilde{Y}_{ij}$  and  $\tilde{Y}_j = \sum_{i \in P} \tilde{Y}_{ij}$  where  $P$  defines the parent space, e.g., at the root

node the parent space is  $P = \{k : k = 1, 2, \dots, 40\}$  for both  $i$  and  $j$  ( $\tilde{\Lambda}_i$  and  $\tilde{\Lambda}_j$  are similarly defined)

1. **Latitudinal (row oriented) partitions:** Calculate  $r_k = \sqrt{\sum_{i=1}^k \tilde{Y}_i} - \sqrt{\sum_{i=1}^k \tilde{\Lambda}_i}$  and  $r_{\sim k} = \sqrt{\sum_{i=k+1}^{40} \tilde{Y}_i} - \sqrt{\sum_{i=k+1}^{40} \tilde{\Lambda}_i}$  for  $k = 1, 2, \dots, 39$ .
2. **Longitudinal (column oriented) partitions:** Calculate  $c_\ell = \sqrt{\sum_{j=1}^\ell \tilde{Y}_j} - \sqrt{\sum_{j=1}^\ell \tilde{\Lambda}_j}$  or  $c_{\sim \ell} = \sqrt{\sum_{j=\ell+1}^{40} \tilde{Y}_j} - \sqrt{\sum_{j=\ell+1}^{40} \tilde{\Lambda}_j}$  for  $\ell = 1, 2, \dots, 39$ .
3. **Select the best partition:** Partition on
  - a. Row 1 to  $k$  and  $k+1$  to 40 if  $\max(r_k, r_{\sim k}) > \max(c_\ell, c_{\sim \ell})$  for all  $\ell$  and  $\max(r_k, r_{\sim k}) \geq \max(r_i, r_{\sim i})$  for all  $i \neq k$ .
  - b. Column 1 to  $\ell$  and  $\ell+1$  to 40 if  $\max(c_\ell, c_{\sim \ell}) > \max(r_k, r_{\sim k})$  for all  $k$  and  $\max(c_\ell, c_{\sim \ell}) \geq \max(c_j, c_{\sim j})$  for all  $j \neq \ell$ .
4. Define  $z_1 = \max(c_\ell, c_{\sim \ell}, r_k, r_{\sim k})$  as partition score for the right-hand partition and let the other offspring have signal-to-noise ratio  $z_{\sim 1}$ . If  $z_1 = c_\ell$  then  $z_{\sim 1} = c_\ell$ ; if  $z_1 = c_{\sim \ell}$  then  $z_{\sim 1} = c_{\sim \ell}$ ; if  $z_1 = r_k$  then  $z_{\sim 1} = r_k$ ; and if  $z_1 = r_{\sim k}$  then  $z_{\sim 1} = r_{\sim k}$ .

Now we repeat the process for the next generation of offspring by considering each offspring of the partition above as a parent space made up of  $p$  rows and  $q$  columns defined by rows  $i_1, i_1+1, \dots, i_p$  and columns  $j_1, j_1+1, \dots, j_q$ . Then a further two offspring are generated by the following steps:

1. **Latitudinal partitions:** Calculate  $r_k = \sqrt{\sum_{i=i_1}^k \tilde{Y}_i} - \sqrt{\sum_{i=i_1}^k \tilde{\Lambda}_i}$  and  $r_{\sim k} = \sqrt{\sum_{i=k+1}^{i_p} \tilde{Y}_i} - \sqrt{\sum_{i=k+1}^{i_p} \tilde{\Lambda}_i}$  for  $k = i_1, i_1+1, \dots, i_p-1$ .
2. **Longitudinal partitions:** Calculate  $c_\ell = \sqrt{\sum_{j=j_1}^\ell \tilde{Y}_j} - \sqrt{\sum_{j=j_1}^\ell \tilde{\Lambda}_j}$  or  $c_{\sim \ell} = \sqrt{\sum_{j=\ell+1}^{j_q} \tilde{Y}_j} - \sqrt{\sum_{j=\ell+1}^{j_q} \tilde{\Lambda}_j}$  for  $\ell = j_1, j_1+1, \dots, j_q-1$ .
3. **Select the best partition:** Partition on:
  - a. Row  $i_1$  to  $k$  and  $k+1$  to  $i_p$  if  $\max(r_k, r_{\sim k}) > \max(c_\ell, c_{\sim \ell})$  for all  $\ell$  and  $\max(r_k, r_{\sim k}) \geq \max(r_i, r_{\sim i})$  for all  $i \neq k \in (i_1, i_1+1, \dots, i_p)$ .
  - b. Column  $j_1$  to  $\ell$  and  $\ell+1$  to  $j_q$  if  $\max(c_\ell, c_{\sim \ell}) > \max(r_k, r_{\sim k})$  for all  $k$  and  $\max(c_\ell, c_{\sim \ell}) \geq \max(c_j, c_{\sim j})$  for all  $j \neq \ell \in (j_1, j_1+1, \dots, j_q)$ .
4. Define  $z_g = \max(c_\ell, c_{\sim \ell}, r_k, r_{\sim k})$  as partition score for the right-hand partition and let the other offspring have signal-to-noise ratio  $z_{\sim g}$ .

The process is repeated for each new generation and only stops when the smooth counts are either

- Too small to alarm an unusually high count (this is discussed in more detail later).
- All cells in the parent space are less than expected.



- The best longitudinal partition and best latitudinal partition are both the parent space.

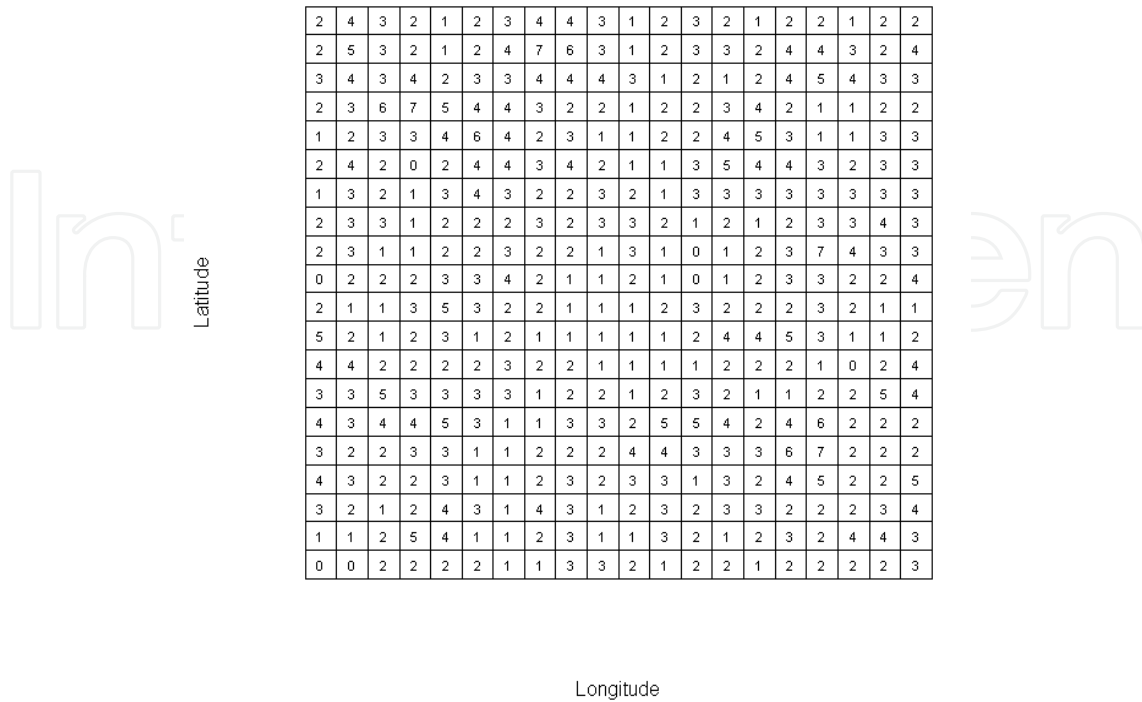


Fig. 3. An example of in-control counts when cell means are correlated.

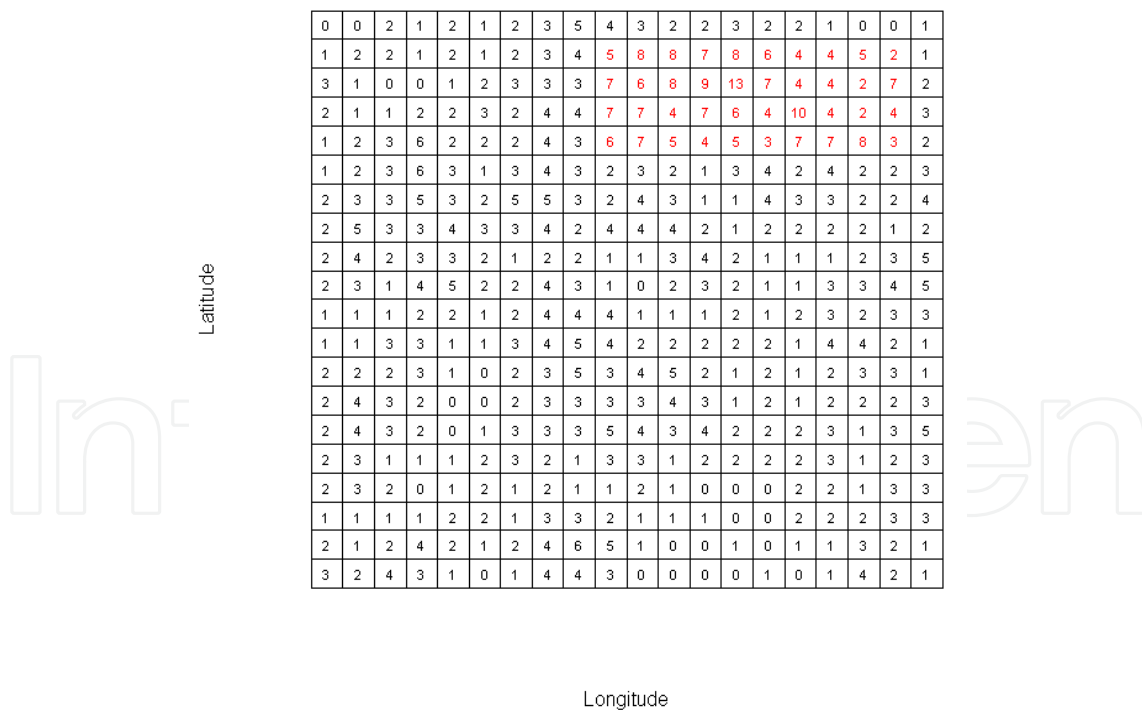


Fig. 4. An example of a hypothetical bio-terrorism event.

An example, of the recursive partitioning when there is no smoothing of counts is now used to demonstrate the growing process for a 20×20 grid. Assume that the in-control Poisson counts for each cell in the grid (except boundary cells) have a mean of 2,5. An example of in-

control counts is presented in Figure 3. Latitude (La) cells are numbered from the bottom of row as 1 to the top row as 20. Longitude (Lo) cells are numbered 1 (left) to 20 (right). Correlated cell counts are noted from Figure 3 by higher than expected cell counts (greater than 2,5) clumping together even when in-control, e.g., latitude number rows 18, 19 and 20 (La>17) and longitude columns 7, 8, 9 and 10 (6<Lo<11). This makes it more difficult to identify bio-terrorism events.

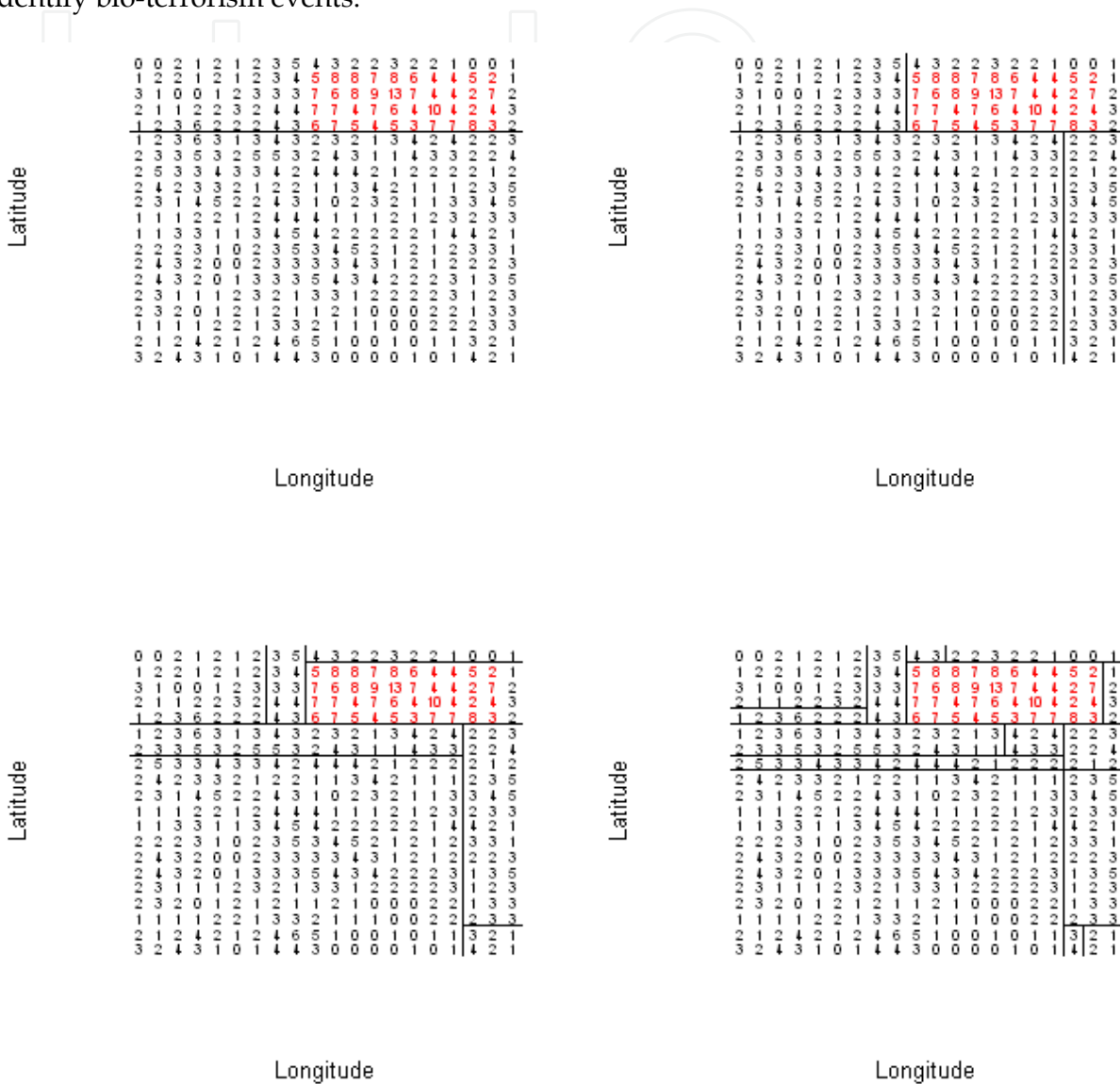


Fig. 5. Presenting offspring for 5 generations using recursive partitioning

We use transitional Poisson regression models to forecast the expected counts one day ahead. We assume that these forecasts, when no bio-terrorism event is present, produce unbiased estimates of the cell means. Therefore, in our application, the in-control count departures from the forecasts are assumed uncorrelated because the dependence is removed by the model forecasts, and the forecast errors are assumed spatially uncorrelated when in-control. The assumption here is that all the spatial correlation is in the mean of the Poisson counts, and the day ahead forecast is an unbiased estimate of this mean. As such, the differences between functions of counts and forecasts will be assumed to be uncorrelated.

Even if this is not true; provided the bootstrap population used to estimate the threshold is accurate, the simulation process will produce an unbiased estimate of the threshold (see Sparks et al. 2010b).

Figure 4 presents another representation of counts illustrated in Figure 3, except in Figure 4 a bio-terrorism event is simulated by adding additional (out-of-control) counts at Latitudes 16, 17, 18 and 19 ( $15 < La < 20$ ) and longitude 10, 11, ..., 19 ( $10 < Lo < 20$ ). The additional counts were assumed to have a mean of 2 per cell. Red is used to highlight the simulated bioterrorism outbreak region in Figure 4. It is clear that the counts in red are higher than the counts for any other part of the  $20 \times 20$  target region. We now demonstrate the recursive partitioning process in trying to find this outbreak.

The recursive partitioning starts by searching for the best partition of longitude and latitude in terms of maximizing the departure of the region's total counts from their expected value, i.e., maximising  $r_k$  and  $c_\ell$  values, respectively. These are defined by rules  $La > 15$  and  $Lo > 9$ , respectively. Generally the region with the greatest fraction of red will be the one with the highest departure from expected and this is true in this case. The best first partition of  $La > 15$  is represented by the black line in the top left-hand table in Figure 5. This first generation of offspring become parents for the next generation of offspring. Each region now grows two offspring by finding the partition that is best for each parent. For the parent region with  $La > 15$ , it is clear that the best partition is  $Lo > 9$  (see the black line in top right-hand table of Figure 5 that separates the red counts from the black counts as best as possible). For the region with  $La < 16$ , the best partition is less obvious - it turns out to be  $Lo > 17$ . The bottom left-hand table in Figure 5 gives the next generation of offspring, and the bottom right-hand table in Figure 5 gives the generation that completely specifies the simulated outbreak. The z-score for the red region is  $z_g = 10,6$  which is higher than any other partitioned region in the table of Figure 5. An exhaustive search of all regions of all different shapes and sizes in

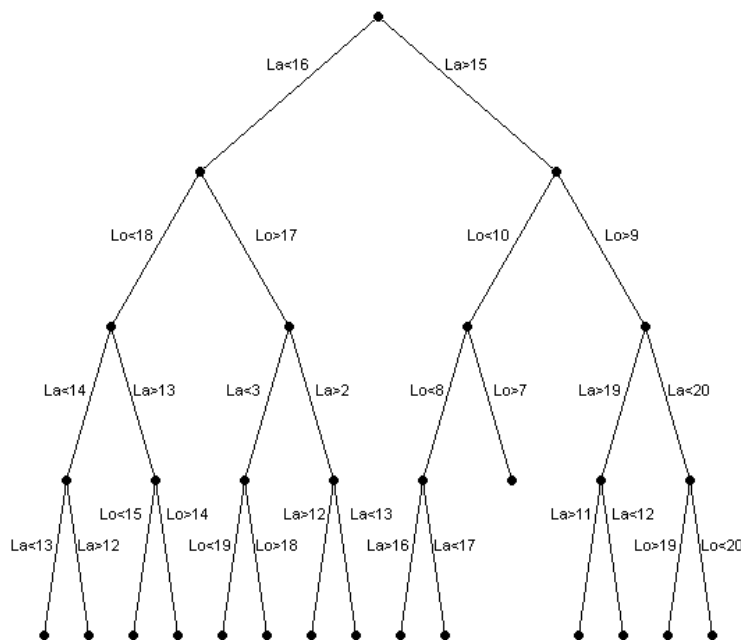


Fig. 6. Surveillance tree representation of the recursive partitions in Figure 5.

Figure 5 revealed that this gave the highest z-score of target region. Recursive partitioning is like stepwise variable selection when building parsimonious regression models – it does not always produce the same result as an exhaustive search. In this example, we have demonstrated that recursive partitioning found the largest outbreak area using significantly smaller number of computations than for example the SCAN statistic with multiple size windows. This balance between computational and detection efficiency is the most attractive feature of the recursive partitioning approach besides its ability to adapt with size and shape of the outbreak.

Figure 6 is a tree based representation of the recursive partitions made in Figure 5. Note that the terminal node defined by rules  $La > 15$ ,  $Lo < 10$  and  $Lo > 7$  does not have any offspring. The  $z_g$  for this node is  $(\sqrt{36} - \sqrt{25}) = 1$  where  $36 = 3 + 5 + 3 + 4 + 3 + 3 + 4 + 4 + 4 + 3$  and  $25 = 2,5 \times 10$ . No partition of this space produces a higher z-score than the parent node's z-score of 1. The closest to this is  $(\sqrt{5 + 4 + 3 + 4 + 3} - \sqrt{5 \times 2,5}) = 0,825$ . Ideally we would like to stop recursive partitioning at this node. In practice, the decision to stop unnecessarily growing of the tree is difficult. Generally we would keep partitioning until some stopping rule applies.

Once partitioning has stopped, then recursive pruning of the terminal nodes commences. The pruning process is outlined in the next section.

### 3.3 Pruning the surveillance tree to deliver a particular false alarm rate

The aim of pruning is to recursively trim away all terminal nodes with smoothed counts that do not differ significantly from their expected values. If all nodes in the tree are pruned away to leave no tree, then no outbreak is signalled. However, if terminal nodes remain after pruning, then an alarm is given. The geographical location of the outbreak is diagnosed by the set of partitioning rules for the remaining terminal nodes. This could involve several non-overlapping locations.

Pruning only starts after growing the tree has stopped. The pruning process is very simple. We prune terminal nodes recursively starting with the last generation offspring in the tree, and then repeat the process recursively with each remaining generation. Only branches with significantly higher than expected counts survive pruning. We prune the least significant offspring if  $z_{\sim g} < h_z$  and prune the generation if  $z_g < h_z$  where  $h_z$  is a positive constant designed to deliver a specified false alarm. Pruning is used to control the false alarm rate measured by either specifying an in-control Average Run Length (ARL) or an in-control Recurrence Interval (RI). The Recurrence Interval (RI) is used to indicate the steady state false alarm rate. ARL and RI are defined more clearly in the simulation section (see Section 4). Pruning starts with the highest values of  $g$  (generations) and stops when either all generations are pruned or when at least one generation (terminal node) survives the pruning process.

The pruning process leaves only generations with smoothed counts significantly higher than expected, e.g., if we started pruning from generation 4 in Figure 5 then only two offspring containing the outbreak regions plus their parents, grandparents, etc would survive the pruning process. If pruning was applied to a tree for the data in Figure 3 with

no outbreaks, then each generation in turn would be pruned away until no offspring remain. Even the target region (the original parent) would be pruned indicating no significant outbreak.

### 3.4 Using the example to demonstrate the stopping rules for the tree growing process

It is useful to stop partitioning when it is known that the threshold will not be exceeded with further partitions. Assume that the threshold is  $h_z = 4,45$  then the node defined by rules  $La > 15$ ,  $Lo < 10$  and  $Lo > 7$  should be terminal because  $(\sqrt{36} - \sqrt{2,5}) = 4,4$ . No partition of this parent space is going to be greater than the threshold of 4,45. Two other nodes would terminate in generation three Figure 5 using  $h_z = 4,45$ . These would be:

- a.  $Lo > 9$  and  $La > 19$  where

$$(\sqrt{4+3+2+2+3+2+2+1+0+0+1} - \sqrt{2,5}) = 2,9 < 4,45.$$

- b.  $La < 3$  and  $Lo > 17$  where

$$(\sqrt{3+2+1+4+2+1} - \sqrt{2,5}) = 2,0 < 4,45.$$

Such a stopping rule stops the unnecessary growing of the surveillance tree, because all offspring will be pruned later. Generally this rule can be defined as stopping when the square root of the total of cell counts for a region is smaller than the threshold plus square root of the smallest cell mean in the region. If the threshold is quite low, then having a threshold on the lower limit for the smoothed count is sensible. Assume that our threshold turns out to be  $h_z = 1,15$  then the above stopping rule does not help much in reducing the partitioning complexity. In the above example, it is sometimes helpful to specify when you would fail to respond, e.g., no response is made for any outbreak with counts less than 5 above expected per day. We would then stop if smoothed counts were below 7,5.

### 3.5 Lattice with cell means that are spatially non-homogeneous

Until now, we have assumed that all cells follow the same model. What if the same time trends persist across cells, e.g., the seasonal trend remains the same for each cell but the mean counts change from cell to cell? How does this influence the threshold for when the model is not homogeneous across all cells? Here we examined the in-control ARL and RI properties as we move the cell mean rate from an average of 0,7 to the current value used of 1,2 by examining its in-control ARL and RI using the same threshold of  $h_z = 1,15$ . The results are recorded in the Table 1 for counts generated using the R function in Appendix A with overall mean  $\mu = 0,1$ , day-of-the-week influences given by  $b_{weekday} = (-0,025, -0,025, -0,025, -0,025, -0,025, 0,05, 0,075)$ , seasonal influences given by  $b_{cos} = 0,1$ ,  $b_{sin} = -0,1$ , transitional influences of  $b_{lags} = (0,1, 0,06, 0,02, 0,01)$ , and  $k = -4$ .

This plan is also reasonably robust to changes in spatial non-homogeneity in means whereas other plans are not. From Table 1 it is clear that the in-control RI values are reasonably stable across a range of mean values. If these mean changes are not systematic as in table above,

but random in space then these changes are expected to be ironed out by the spatial smoothing. Adaptive plans can be derived using the results in Table 1 based on Sparks (2000) and Sparks et al (2010a).

Average counts	0,7	0,8	0,9	1,0	1,1	1,27
$h_z$	1,15	1,15	1,15	1,15	1,15	1,15
	1,2	1,2	1,2	1,2	1,2	1,2
	1,25	1,25	1,25	1,25	1,25	1,25
In-control ARL	55,4	62,5	73,7	78,7	83,1	88,7
	72,4	84,4	100,2	106,4	109,7	117,9
	81,1	91,8	111,4	119,2	126,2	149,3
In-control RI	50,3	51,6	52,3	56,1	60,1	69,3
	61,0	63,9	68,7	72,4	68,7	82,5
	75,0	83,6	86,5	88,9	86,5	97,4

Table 1. In-control ARLs and RIs for specified thresholds of 1,15, 1,2 and 1,25

#### 4. Simulation study

This section provides the results of simulation studies comparing the performance of the SCAN statistic with that of the EWMA Surveillance Trees for two outbreak shapes and locations using Average Time to Signal and Average Run Length as the measures of performance. When the size and shape of the scanning window matches that of the outbreak in the SCAN plan, then it remains optimal. However, the flexible partitions of EWMA Surveillance Trees result in robust performance across a range of outbreak dimensions, without being optimal for any specific outbreak size or shape.

We simulate uncorrelated Poisson counts on a 40x40 spatial grid using the R function in Appendix A as described in Section 3.4, with the exception that the mean value is bounded at a minimum of one (i.e.,  $k=0$ ) which makes the counts more homogeneous. Denote these counts by  $\bar{c}_{i,j}$  for row  $i$  and column  $j$ . These counts are then aggregated locally as follows for non-boundary cells:

$$c_{i,j} = \text{floor}(\bar{c}_{i,j} + 0,3 \times (\bar{c}_{i-1,j} + \bar{c}_{i,j-1} + \bar{c}_{i+1,j} + \bar{c}_{i,j+1}) + 0,1 \times (\bar{c}_{i-1,j-1} + \bar{c}_{i-1,j+1} + \bar{c}_{i+1,j-1} + \bar{c}_{i+1,j+1}))$$

where floor means round down to the nearest whole number. Boundary cells have smaller means and their counts are defined as in Table 2.

This generates more realistic spatially correlated in-control counts for infectious or contagious diseases. Note that these counts are not expected to be Poisson distributed, but tend to be under-dispersed counts relative to the Poisson distribution. We assume no knowledge of this under-dispersion in trying to forecast counts. We model the trends in the counts using transitional Poisson regression model and assume that if the counts are in-control then the one day ahead forecast errors are uncorrelated. In situations where spatial correlation persists in the forecast errors for in-control situations, an alternative approach is models which account for this spatial correlation, such as seeming unrelated Poisson regression models (King, 1989, and Grijalva, T., Bohara, *et al.* 2003), and multivariate Poisson

regression model (Karlis and Meligkotsidou, 2005, and Bermudez and Karlis, 2011). However, here we model each of the 1600 cells separately using Poisson regression models very similar to that used in Sparks *et al.* (2010a) and Sparks *et al.* (2011a) that use explanatory variables: logarithm of lag counts plus 1, day-of-the-week, public holidays and harmonics. These models are used to produce one-day ahead forecasts which estimate the “Poisson” means for the next day.

$i = 1, 1 < j < 40$	$c_{1,j} = \text{floor}(\bar{c}_{1,j} + 0,3 \times (\bar{c}_{1,j+1} + \bar{c}_{1,j-1} + \bar{c}_{2,j}) + 0,1 \times (\bar{c}_{2,j-1} + \bar{c}_{2,j+1}))$
$i = 40, 1 < j < 40$	$c_{40,j} = \text{floor}(\bar{c}_{1,j} + 0,3 \times (\bar{c}_{40,j+1} + \bar{c}_{40,j-1} + \bar{c}_{39,j}) + 0,1 \times (\bar{c}_{39,j-1} + \bar{c}_{39,j+1}))$
$1 < i < 40, j = 1$	$c_{i,1} = \text{floor}(\bar{c}_{i,1} + 0,3 \times (\bar{c}_{i+1,1} + \bar{c}_{i-1,1} + \bar{c}_{i,2}) + 0,1 \times (\bar{c}_{i-1,2} + \bar{c}_{i+1,2}))$
$1 < j < 40, j = 40$	$c_{i,40} = \text{floor}(\bar{c}_{i,1} + 0,3 \times (\bar{c}_{i+1,40} + \bar{c}_{i-1,40} + \bar{c}_{i,39}) + 0,1 \times (\bar{c}_{i-1,39} + \bar{c}_{i+1,39}))$
$i = 1, j = 1$	$c_{1,1} = \text{floor}(\bar{c}_{1,1} + 0,3 \times (\bar{c}_{2,1} + \bar{c}_{1,2}) + 0,1 \times \bar{c}_{2,2})$
$i = 40, j = 40$	$c_{40,40} = \text{floor}(\bar{c}_{40,40} + 0,3 \times (\bar{c}_{39,40} + \bar{c}_{40,39}) + 0,1 \times \bar{c}_{39,39})$

Table 2. The process used to simulate cells counts that are spatially correlated.

The recursive partitioning process is then applied to the temporal or spatio-temporal smoothed counts. The  $z_g$  statistic is used to assess the departure of smoothed counts from smoothed day ahead forecasted counts. When in-control,  $z_g$  has mean zero and approximately a constant variance. The convenience of the statistic is discussed in Sparks *et al.* (2010b) and has been demonstrated to compare very well with the SCAN statistic.

The simulation process generated in-control counts. Simulated bio-terrorism outbreaks are generated by adding to these in-control counts additional Poisson counts for a fixed rectangular region. The outbreak region is then hidden and we examined how early the plans alarm this outbreak. Rectangular outbreak regions were generated involving either 10x4 cells or 20x2 cells. Zero-state in-control ARLs and the Recurrent Interval (RI) were estimated as follows:

- Start the Run Length (RL) at 1. Generate in-control cell counts for the first day, and perform the surveillance tree process. If no tree results after pruning then go to the next step.
- Increase the RL by 1. Generate in-control cell counts for the next day, perform the surveillance tree process, and if no tree results after pruning, then go to the next step.
- Repeat the previous step until some branches of the surveillance tree survive the pruning process. Record the RL (denote this  $RL_1$ ) and go to the next step.
- Generating in-control cell counts for the several consecutive days until there are at least 7 consecutive days with no tree surviving, then set  $RI=7$ .
- Increase the RI by 1. Generate in-control cell counts for the next day, perform the surveillance process, and if no tree results after pruning, then go to the next step.
- Repeat the previous step until some branches of the surveillance tree survive the pruning process. Record the RI (denote this  $RI_1$ ) and go to the next step.
- Repeat all steps above 1600 times to give  $(RL_i, RI_i), i = 1, 2, \dots, 1600$ . Average these to give the in-control ARL and an estimate of the expected RI.

The 1600 simulations were selected balancing both computational effort and estimation accuracy. Traditionally, the standard deviation for RL with an in-control ARL=100 is about 100. The standard deviation from the in-control ARL is expected to be roughly  $100 / \sqrt{1600} = 2,5$ .

The out-of-control ARL are calculated in the same way as the in-control ARL, except with each run additional counts with the same mean are added to the outbreak region. The results are reported in Table 1. The recurrence interval for out-of-control situations was not very interesting because generally when a plan signalled it did not stop signalling. The out-of-control ARLs are therefore the zero state out-of-control ARLs. Reporting the in-control RI values provides readers insight into the steady state in-control performance.

Outbreak	Recursive partitioning with temporal smoothing ( $\alpha = 0,1$ ) $h_z = 1,255$		Recursive partitioning with temporal smoothing ( $\alpha = 0,1$ ) $h_z = 1,404$		Recursive partitioning with spatio-temporal smoothing ( $\lambda = 0,7, \alpha = 0,1$ ) $h_z = 1,15$	
	Rows	Columns	Rows	Columns	Rows	Columns
ARL/RI $\delta$	108,9 / 37,5	108,9 / 37,5	201,5/70,6	201,5/70,6	108,3 / 69,6	108,3 / 69,6
1	<b>11,88</b>	<b>11,95</b>	13,25	12,00	15,42	16,26
2	6,78	<b>4,86</b>	7,96	4,91	<b>5,61</b>	6,39
3	4,51	<b>3,55</b>	5,12	4,11	<b>3,94</b>	3,95
4	3,72	<b>2,76</b>	3,80	3,20	<b>2,94</b>	2,93
5	3,02	<b>2,35</b>	3,24	2,55	<b>2,65</b>	2,44
6	2,51	2,12	2,72	2,19	<b>2,10</b>	<b>2,11</b>
7	2,44	<b>1,70</b>	2,50	1,77	<b>1,97</b>	1,79
8	2,40	1,55	2,41	1,60	<b>1,70</b>	<b>1,53</b>
9	2,34	<b>1,45</b>	2,36	1,50	<b>1,56</b>	<b>1,45</b>
10	2,10	1,31	2,12	1,32	<b>1,20</b>	<b>1,30</b>
11	2,02	<b>1,12</b>	2,04	1,16	<b>1,15</b>	1,16
12	1,91	<b>1,05</b>	1,91	1,07	<b>1,04</b>	1,08

Table 3. ARL performance of the surveillance tree plans when the outbreak spans regions of 40 cells  $\delta$  is the increase in mean for cells in the outbreak region.

Recursive partitioning does well when the outbreak is located on the boundary of the target region, partially because fewer partitions are necessary to find it. Another reason is that the boundary cells are where counts are smoothed less and the mean rates are lowest on the boundary, and therefore the signal is the highest. In other words, boundary outbreaks are easier to find using recursive partitioning because fewer generations are needed to find them and the signal-to-noise ratio on the boundary in this simulation process is higher.

The robust performance of plans, i.e., when nothing is known about the shape and size of the outbreak, is an important consideration because in bio-terrorism nothing is known about the size or shape of the outbreak. Therefore, Table 3 looks at a variation in rectangular shaped outbreaks. The recursive partitioning plan's flexibility in partitions strengthens its robust performance across a range of outbreak sizes (see Sparks *et al.* 2011c). In other words,



if the outbreak dimensions are known, then the SCAN plan can always be trained to be more efficient than the recursive partitioning plan, and therefore it is preferred in these circumstances. But when nothing is known about its size and shape recursive partitioning is a more flexible and robust approach than the SCAN plan.

Table 3 demonstrates that spatial smoothing prior to applying recursive partitioning of temporally smoothed counts has early detection advantages for the same in-control RI values of approximately 70, e.g., for large shift in mean counts of 12 per cell the ARL is 1,04 as opposed to 1,91 (almost a day later). The spatial smoothing also closes the gap between the in-control ARL and in-control RI, and therefore it is likely to have equivalent steady state early detection advantages.

The future research challenges are to investigate what levels of spatial smoothing are best for early detection. In these studies we only tried  $\lambda = 0,7$ . In addition, there may be an interaction between the temporal smoothing and spatial smoothing. Therefore this needs to be investigated in future work.

The SCAN plan was applied using two separate scanning window sizes: 5 by 5 cells and 10 by 10 cells. As nothing is known about future outbreaks, therefore we consider square  $m$  by  $m$  scans – for at least the first outbreak considered in Table 4 the 5 by 5 cells scans is close to the  $m$  by  $m$  scanning plan with the smallest out-of-control ARLs. Unlike the recursive partitioning for the scan plans the expected values were taken as known, and therefore this gives an advantage to the SCAN plan in this comparison. Note that there was less of a difference between the in-control ARL and in-control RI for the scan plans than the surveillance tree plan. This mostly occurs because of the way we estimated the RI. In addition, by not allowing a signal in the first 7 values after a outbreak signal means that there are only a correlation of 0,3 between the signalled time point and the start of the RI for the scan plan, whereas in plans using the temporal EWMA smoothing with  $\alpha = 0,1$  this is 0,48. Hence the way we estimated RI is going to be more influenced by the previous signal in the surveillance tree plan and therefore its RI is smaller. Clearly, the more the outbreak matches the scanning region the earlier the SCAN plan detects outbreaks, but it is clear that it lacks the flexibility in terms of detecting outbreaks of unknown shape and size. It is worth noting that when the SCAN plan does perform better than the surveillance trees (comparing Tables 3 and 4), the surveillance trees are not far behind in performance whereas the reverse is not true. Figure 7 provides a graphical comparison of the ARL properties. From Figure 7, the SCAN plan 5 by 5 by 10 days is best on average for detecting a 10 by 4 outbreak which is near the centre of the target region but significantly worse for the 2 by 20 outbreak. The 2 by 20 outbreak is quite feasible, e.g., this shaped outbreak would result if the outbreak was started in railway carriages travelling along a specific route.

## 5. Conclusions

The recursive partitioning approach can easily be extended to deal with more than two or three dimensions. It is ideal for finding disease outbreaks which cluster in multivariate space that could also include variables such as triage category (severity), age or gender, whereas the SCAN plans become computationally infeasible in higher dimensions. However further research is required to determine whether we can extend the benefit found from spatial smoothing to other dimensions that are not spatial, for example smoothing over age groups.

An advantage of the ability to extend to higher dimensions is that Surveillance Trees could be used to monitor several related diseases or symptoms jointly using disease category as a variable searched over in the recursive partitioning process. This would mean having one false alarm rate for all disease groups being monitored. Such planning may be beneficial in terms of managing the effort an epidemiological unit can devote to investigating false alarms. Similarly the method could also be used for syndromic surveillance - the surveillance of public health records such as Emergency Department visits. Detecting increases in ill-defined syndromes that may indicate the use of new and unknown biological agents.

Because of the efficiency of computation, another advantage of recursive partitioning is that it could be implemented using a hand-held mini-computer in the field to assess whether say counts have increased more than expected for any cluster of a target region. It does not need a sophisticated computer, as opposed to the SCAN plan.

Ultimately the EWMA Surveillance Tree methodology is a promising tool for detecting spatio-temporal outbreaks when the size and location of the outbreak is unknown. By incorporating the benefits of temporal and spatial smoothing with a tool for efficiently locating outbreaks, the method is well suited for the surveillance of bioterrorism events.

Outbreak	SCAN plan for 5 by 5 scans Threshold p-value=0,9999999999999991		SCAN plan for 10 by 10 scans Threshold p-value=0,9999999999999994		
	Rows Columns	10:19 16:19	10:11 1:20	10:19 16:19	10:11 1:20
ARL/RI $\delta$		94,5 / 77,2	94,5 / 77,2	80,5/65,6	80,5/65,6
1		14,22	49,66	32,38	59,95
2		5,63	8,14	5,80	15,26
3		<b>3,85</b>	5,31	4,08	5,97
4		<b>2,99</b>	4,39	3,28	4,88
5		<b>1,79</b>	3,55	2,54	3,99
6		<b>1,60</b>	3,07	2,23	3,39
7		<b>1,54</b>	2,63	1,85	2,92
8		<b>1,30</b>	2,30	1,78	2,53
9		<b>1,22</b>	2,09	1,56	2,31
10		<b>1,15</b>	1,98	1,43	2,11
11		<b>1,05</b>	1,78	1,30	1,91
12		<b>1,01</b>	1,68	1,19	1,72

Table 4. ARL performance of the SCAN plan when the outbreak spans regions of 40 cells.

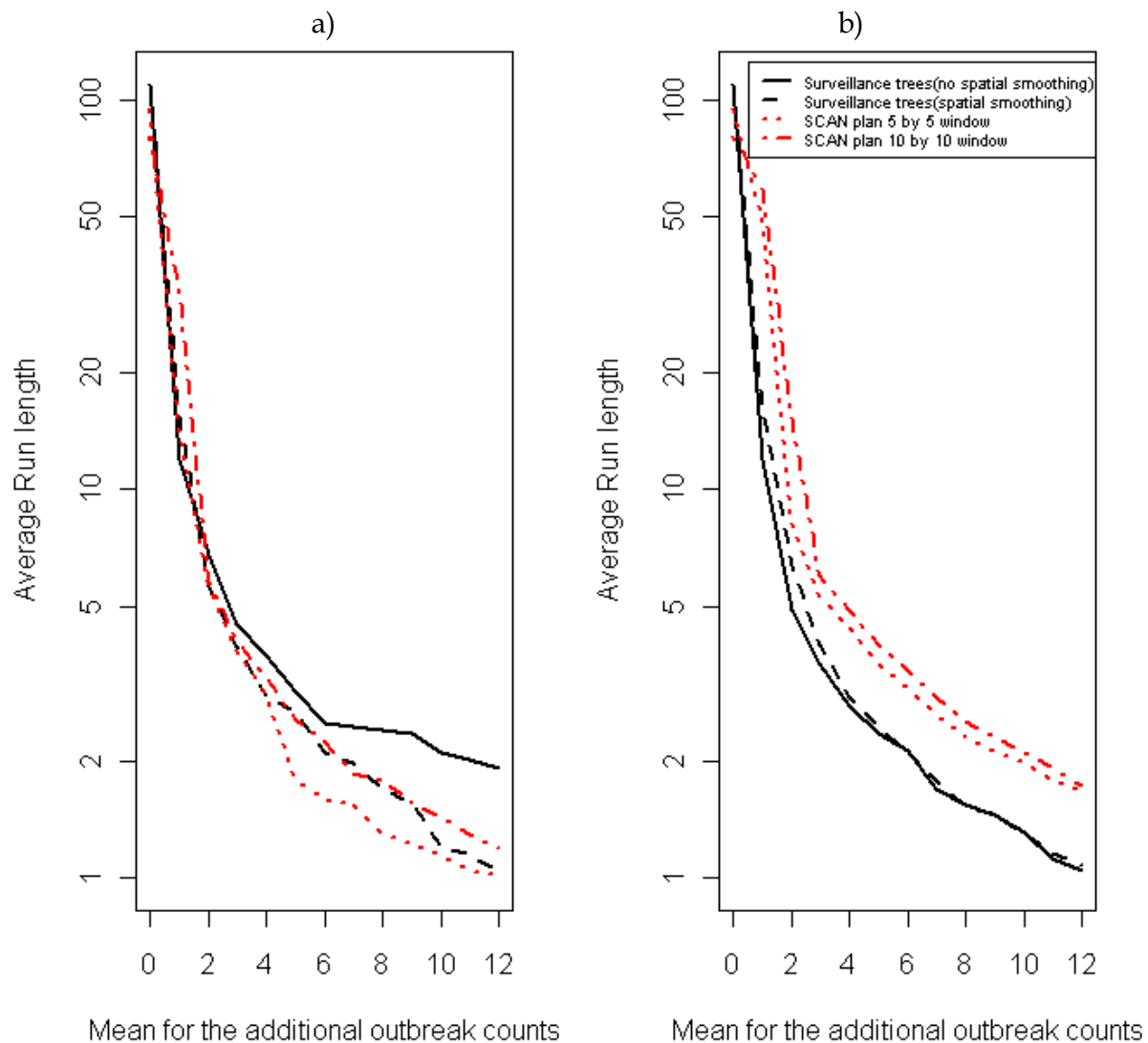


Fig. 7. In-control and out-of-control ARL for the SCAN plan and Surveillance Tree plan

## 6. Appendix

Inputs to the function are  $\mu$ =overall mean;  $b_{\text{weekday}}$ =day-of-the-week influence;  $b_{\text{cos}}$  and  $b_{\text{sin}}$  = influence of the season;  $b_{\text{lags}}$  = is the influence of previous counts;  $n_{\text{ob}}$ = number of counts generated;  $k$  is the threshold specifying the lowest expected count allowed (Poisson counts are truncated at times).

```
transitionalPoisson<-function(mu,bweekday,bcos,bsin,blags,nob,k){
  nlag<-length(blags)
  lcounts<-rep(0,nlag)
  dw<-1
  for(i in 1:nob){
    temp<-
    mu+bweekday[dw]+bcos*cos(2*pi*i/365.25)+bsin*sin(2*pi*i/365.25)+sum(blags*log(lcounts
    +1))
```

```

temp[temp<k]<-k
if(i==1)ncount<-rpois(1,exp(temp)) else ncount<-c(ncount,rpois(1,exp(temp)))
lcounts[2:nlag]<-lcounts[1:(nlag-1)]
lcounts[1]<-ncount[i]
dw<-dw+1
if(dw>7)dw<-1}
ncount}

```

## 7. References

- Bermudez, L. & Karlis, D. (2011). Bayesian multivariate Poisson models for insurance ratemaking. *Insurance: Mathematics and Economics*, 48:226-236.
- Currie, I., Durban, M. & Eilers, P. (2003). Using p-splines to extrapolate two-dimensional poisson data. In *Proceedings of the 18th International Workshop on Statistical Modeling*, Verbeke, G. Molenberghs, A., & Fieuws, S. (Eds.). Katholieke Universiteit Leuven: 97-102.
- Chen, J., & Glaz, J. (1996). Two-Dimensional Discrete SCAN Statistics, *Statistics and Probability Letters*, 31 : 59-68.
- Glaz, J., Naus, J. & Wallenstein, S. (2001) *SCAN Statistics*. Springer, New York.
- Grijalva, T., Bohara, A.K. & Berrens, R.P. (2003). A seemingly unrelated Poisson model for revealed and stated preference data, *Applied Economics Letters*, 10: 443-446.
- Han, S. W., Mei, Y. & Tsui, K.-L. (2008). A comparison between SCAN and CUSUM methods for detecting increases in Poisson rates, *Technical Report*, School of ISyE, Georgia Institute of Technology.
- Joner M. D., Woodall, W.H. Reynolds, M.R., Fricker, R.D. (2008). A One-sided MEWMA Chart for Health Surveillance. *Quality and Reliability Engineering International*. 24: 503-518.
- Karlis, D. & Meligkotsidou, L. (2005). Multivariate Poisson regression with covariance structure. *Statistics and Computing*, 15: 255-265.
- King, G. (1989), "A Seemingly Unrelated Poisson Regression Model," *Sociological Methods and Research*, 17: 235-255.
- Kulldorff, M. & Nagarwalla N. (1995). Spatial disease clusters: Detection and Inference. *Statistics in Medicine*, 14:799-810.
- Kulldorff, M. (1997). A spatial SCAN statistic, *Communications in Statistics: Theory and Methods*, 26:1481-1496.
- Kulldorff, M. (2001). Prospective time periodic geographical disease surveillance using a SCAN statistic, *Journal of the Royal Statistical Society Series A-Statistics in Society* 164: 61-72.
- Kulldorff, M., Heffernan R., Hartman J., Assunção RM., Mostashari F. (2005). A space-time permutation SCAN statistic for the early detection of disease outbreaks, *PLoS Medicine*, 2:216-224.
- Lee, D-J. & Durban, M. (2009). Smooth-CAR mixed models for spatial count data. *Computational Statistics and Data Analysis*, 53: 2968-2979.
- Raubertas, R.F. (1989). An analysis of disease surveillance data that uses the geographic locations of reporting units, *Statistics in Medicine*, 18: 2111-2122.
- Riedel, K.S. (1993). Optimal data-based kernel estimation of evolutionary spectra. *IEEE Transactions on signal processing*, 41: 2439-2447.

- Rogerson, P.A. & Yamada, I. (2004). Monitoring Change in Spatial Patterns of Disease: Comparing Univariate and Multivariate Cumulative Sum Approaches. *Statistics in Medicine*, 23 (14): 2195-2214.
- Rogerson, P.A., Lee, G. & Yamada, I. (2006). Statistical methods for detection and monitoring of spatial clusters. Technical report. See [www.acsu.buffalo.edu/~rogerson/technical%20report.doc](http://www.acsu.buffalo.edu/~rogerson/technical%20report.doc)
- Sarka, III, J.L. (2011). Surveillance of negative binomial and Bernoulli processes. Ph.D. Thesis, Virginia Tech.
- Sparks, R. (2000). CUSUM Charts for Signalling Varying Location Shifts. *Journal of Quality Technology*. 32(2):157-171.
- Sparks, R., Carter, C., Graham, P.L., Muscatello, D., Churches, T., Kaldor, J., Turner, R., Zheng, W. & Ryan, L. (2010a). Understanding sources of variation in syndromic surveillance for early warning of natural or intentional disease outbreaks. *IIE Transactions*, 42(9): 613-631.
- Sparks, R.S. & Okugami, C. (2010b). Surveillance trees: early detection of unusually high number of vehicle crashes, *InterStat*, January, see <http://interstat.statjournals.net/YEAR/2010/abstracts/1001002.php>
- Sparks, R.S., Keighley, T. & Muscatello, D. (2011a). Exponentially weighted moving average plans for detecting unusual negative binomial counts. *IIE Transactions* 42:721-733.
- Sparks, R.S., Keighley, T. & Muscatello, D. (2011b). Optimal exponentially weighted moving average (EWMA) plans for detecting seasonal epidemics when faced with non-homogeneous negative binomial counts. *Journal of Applied Statistics*. DOI: 10.1080/02664763.2010.545184.
- Sparks, R.S. (2011c). Spatially clustered outbreak detection using EWMA SCAN statistics with multiple sized windows. *Communications in Statistics - Simulation and Computation*. To appear.
- Steiner, S.H., Grant, K., Coory, M. & Kelly, H.A. (2011). Detecting the start of an influenza outbreak using exponentially weighted moving average charts. *BMC Medical Informatics and Decision Making*, 10:37.
- Tango, T. (1995). A class of test for detecting general and focused clustering of rare diseases. *Statistics in Medicine*, 14: 2323-23334.
- Woodall, W. H., Marshall, J. B. , Joner, M. D. Jr., Fraker, J. E. & Abdel-Salam, A. G. (2008). On the use and evaluation of prospective SCAN methods for health-related surveillance, *Journal of the Royal Statistical Society: Series A* 171: 223-237.
- Yu, X., Tang, L.A. & Han, J. (2009). Filtering and Refinement: A two stage approach for efficient detection of anomalies. 9th IEEE International Conference on Data Mining.



## **Bioterrorism**

Edited by Dr. Stephen Morse

ISBN 978-953-51-0205-2

Hard cover, 192 pages

**Publisher** InTech

**Published online** 28, March, 2012

**Published in print edition** March, 2012

This book consists of nine chapters, written by international authorities, discussing various aspects of bioterrorism preparedness and response. Five of the chapters are agent-specific and highlight the pathogenesis, prevention and treatment, and the potential of specific organisms (*Rickettsia* and *Yersinia pestis*) or toxins (ricin, botulinum neurotoxins, and staphylococcal enterotoxins) to be used for nefarious purposes. Four chapters discuss different aspects of detecting and responding to a bioterrorism attack. These include methods for spatio-temporal disease surveillance, international laboratory response strategies, detection of botulinum neurotoxins in food and other matrices, and the use of physical methods (ie Raman spectroscopy) to detect spores.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ross Sparks, Sarah Bolt and Chris Okugami (2012). Spatio-Temporal Disease Surveillance, *Bioterrorism*, Dr. Stephen Morse (Ed.), ISBN: 978-953-51-0205-2, InTech, Available from:  
<http://www.intechopen.com/books/bioterrorism/spatio-temporal-disease-surveillance>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen