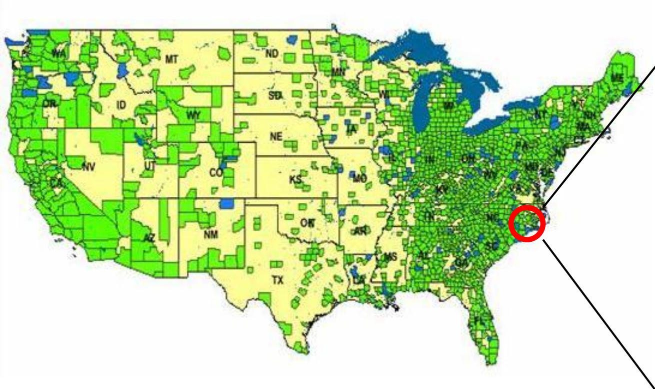
Fast Subset Scanning for Multivariate Event Detection

Daniel B. Neill
Carnegie Mellon University
H.J. Heinz III College

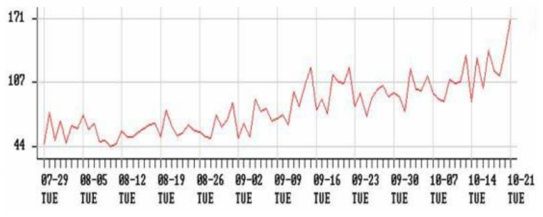
E-mail: neill@cs.cmu.edu

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

Multivariate event detection



Spatial time series data from spatial locations s_i (e.g. zip codes)



Time series of counts $c_{i,m}^t$ for each zip code s_i for each data stream d_m.

Outbreak detection

d_1 = respiratory ED

d_2 = constitutional ED

d_3 = OTC cough/cold

d_4 = OTC anti-fever

(etc.)

Our goals: **detect** any emerging events (e.g. disease outbreaks), **characterize** the type of event, and **pinpoint** the affected locations.

Possible hypotheses

$H_1(S, E_k)$ – event of type E_k has occurred in space-time region S.

H_0 – null hypothesis that no events have occurred.

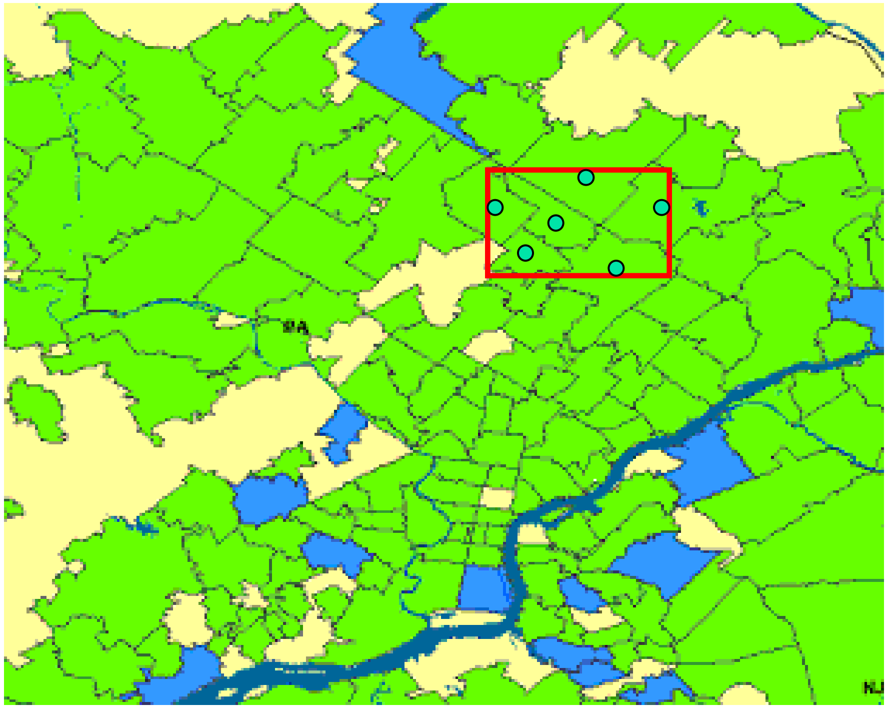
Simplifying assumptions

Univariate, purely spatial case (single event type, single data stream, and single time step)

$H_1(S)$ vs. H_0, $S \subseteq \{s_1..s_N\}$

Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

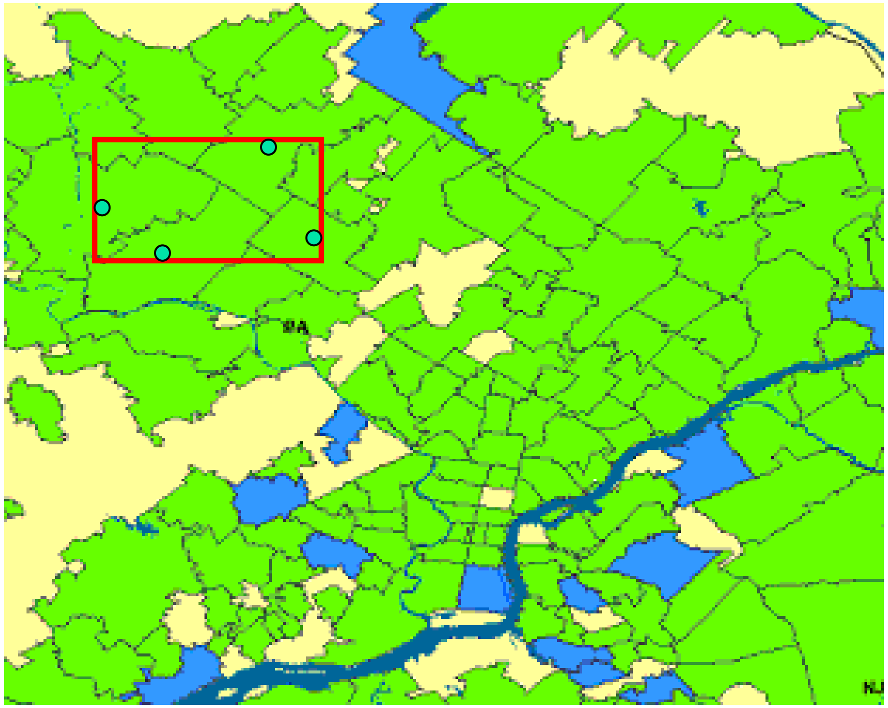


To detect and localize events, we can search for spatial regions where the observed counts are significantly higher than expected.

Imagine moving a spatial window around the scan area, allowing the window size and shape to vary.

Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

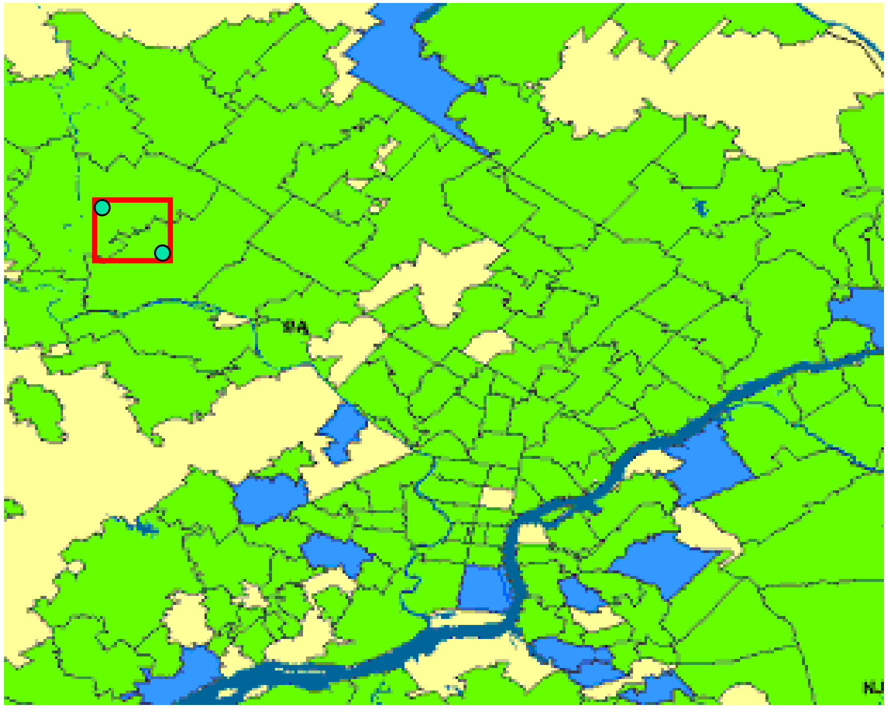


To detect and localize events, we can search for spatial regions where the observed counts are significantly higher than expected.

Imagine moving a spatial window around the scan area, allowing the window size and shape to vary.

Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

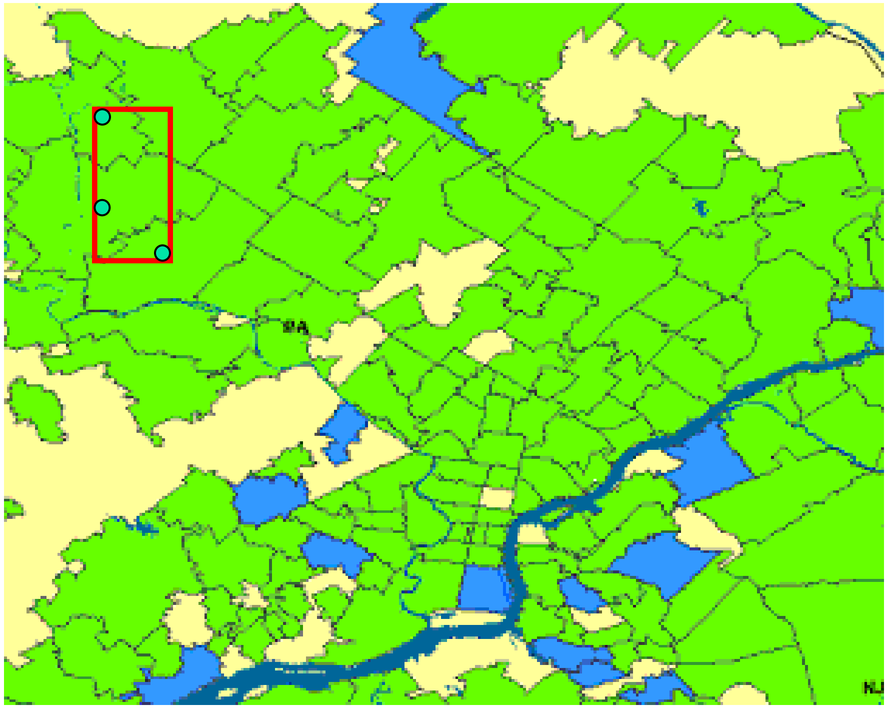


To detect and localize events, we can search for spatial regions where the observed counts are significantly higher than expected.

Imagine moving a spatial window around the scan area, allowing the window size and shape to vary.

Expectation-based scan statistics

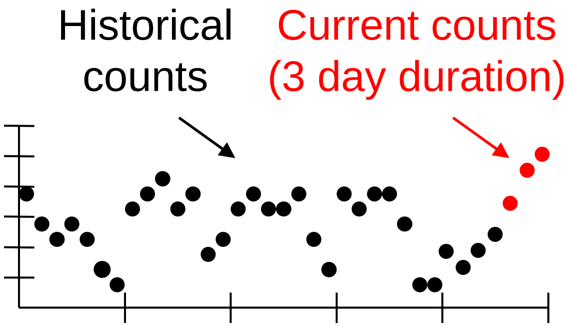
(Kulldorff, 1997; Neill and Moore, 2005)



To detect and localize events, we can search for spatial regions where the observed counts are significantly higher than expected.

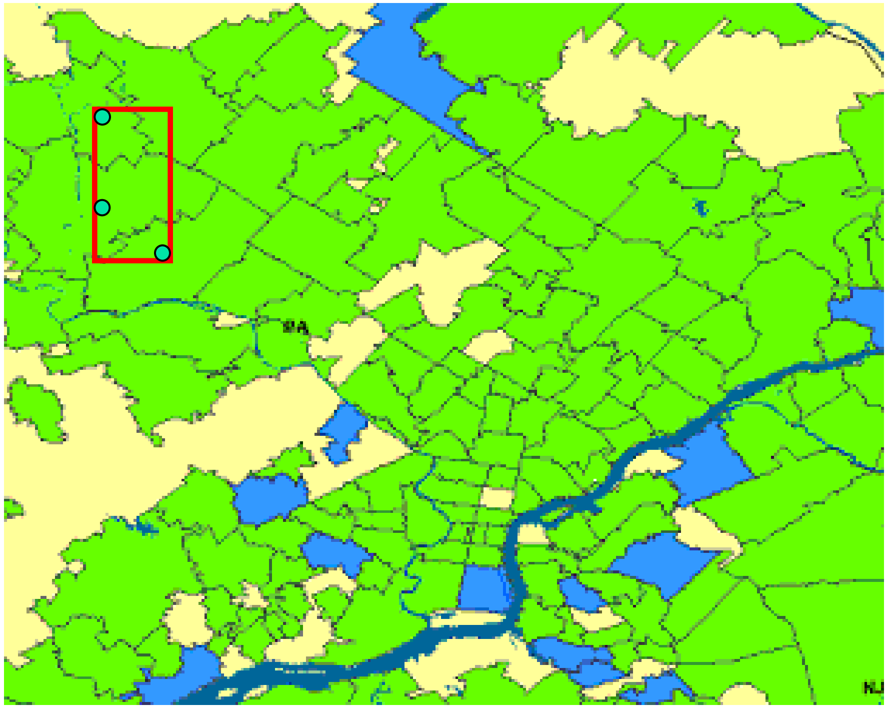
Imagine moving a spatial window around the scan area, allowing the window size and shape to vary.

For each of these regions, we compare the current counts for each location to the time series of historical counts for that location.



Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)



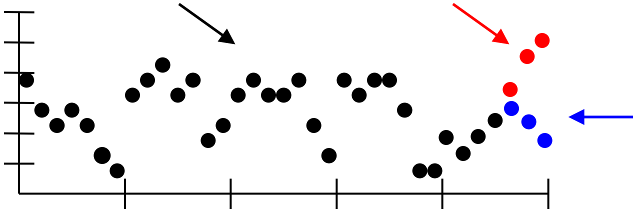
For the standard scan statistic approach, we assume that each count is drawn from a Poisson distribution with unknown mean.

We perform time series analysis to find the <u>expected counts</u> for each recent day, then compare actual to expected counts.

For each of these regions, we compare the current counts for each location to the time series of <u>historical counts</u> for that location.

Historical counts

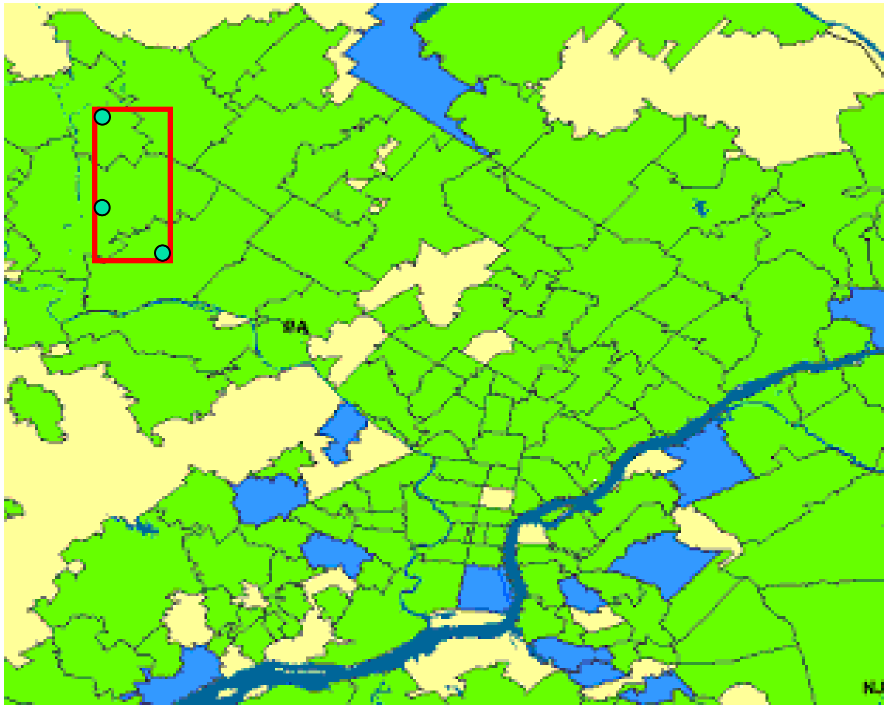
Current counts (3 day duration)



Expected counts

Expectation-based scan statistics

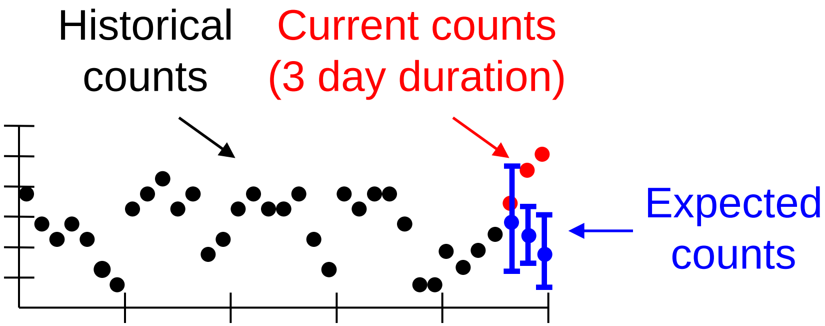
(Kulldorff, 1997; Neill and Moore, 2005)



For the standard scan statistic approach, we assume that each count is drawn from a Poisson distribution with unknown mean.

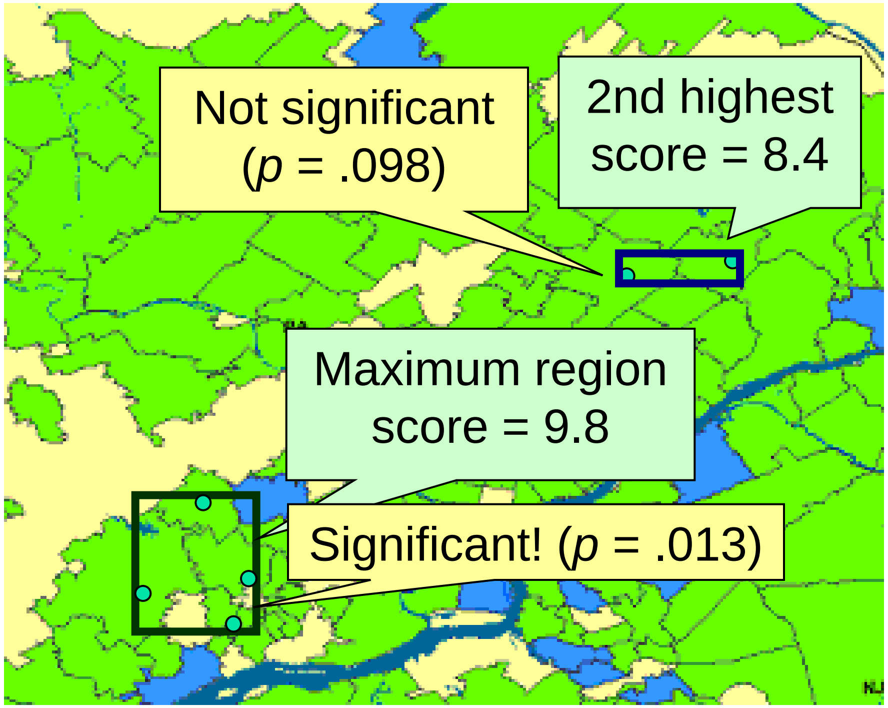
Similarly, we can compute a Gaussian scan statistic by obtaining the expectations and variances from historical data.

For each of these regions, we compare the current counts for each location to the time series of <u>historical counts</u> for that location.



Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)



Not significant (p = .098)

2nd highest score = 8.4

Maximum region score = 9.8

Significant! (p = .013)

We find the regions with highest values of a likelihood ratio statistic, and compute the *p*-value of each region by randomization testing.

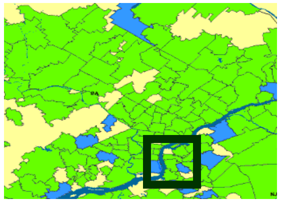
$$F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$$

Alternative hypothesis: outbreak in region S

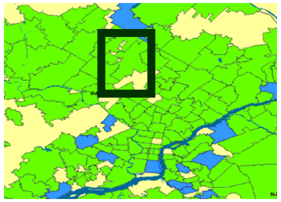
Null hypothesis: no outbreak

To compute p-value
Compare region score to maximum region scores of simulated datasets under H_0.

$F_1^* = 2.4$

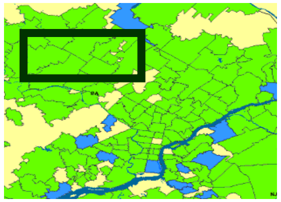


$F_2^* = 9.1$



…

$F_{999}^* = 7.0$



Poisson scan statistic models

Counts are Poisson distributed: $c_i \sim \text{Poisson}(q_i b_i)$

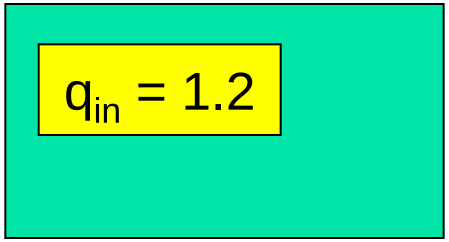
q_i is relative risk, b_i is expected count under H_0

Expectation-based Poisson (EBP)

(Neill and Moore, 2005)

H_0: $q_i = 1$ everywhere
(counts = expected)

$H_1(S)$: $q_i = q_{in}$ in S and $q_i = 1$ outside, for some $q_{in} > 1$.
(counts > expected in S)

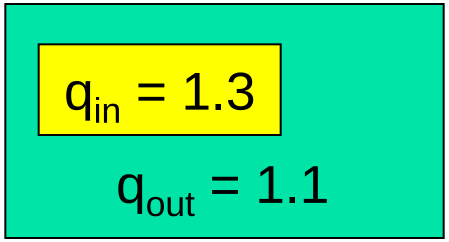


Population-based Poisson (PBP)

(Kulldorff, 1997, 2001)

H_0: $q_i = q_{all}$ everywhere
(inside = outside)

$H_1(S)$: $q_i = q_{in}$ in S and $q_i = q_{out}$ outside, for some $q_{in} > q_{out}$.
(inside > outside)



Poisson scan statistic models

Counts are Poisson distributed: $c_i \sim \text{Poisson}(q_i b_i)$ — q_i is relative risk, b_i is expected count under H_0

Expectation-based Poisson (EBP)

(Neill and Moore, 2005)

H_0: $q_i = 1$ everywhere
(counts = expected)

$H_1(S)$: $q_i = q_{in}$ in S and $q_i = 1$ outside, for some $q_{in} > 1$.
(counts > expected in S)

$$F(S) = \left(\frac{C}{B}\right)^{C} e^{B-C}$$

(if C > B)

Population-based Poisson (PBP)

(Kulldorff, 1997, 2001)

H_0: $q_i = q_{all}$ everywhere
(inside = outside)

$H_1(S)$: $q_i = q_{in}$ in S and $q_i = q_{out}$ outside, for some $q_{in} > q_{out}$.
(inside > outside)

$$F(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{B_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{B_{all}}\right)^{-C_{all}}$$

(if $C_{in} / B_{in} > C_{out} / B_{out}$)

Gaussian scan statistic models

Counts are Gaussian distributed: $c_i \sim \text{Gaussian}(q_i b_i, \sigma_i)$

Let $C' = \Sigma\, c_i' = \Sigma\, c_i b_i / \sigma_i^2$ and $B' = \Sigma\, b_i' = \Sigma\, b_i^2 / \sigma_i^2$.

Expectation-based Gaussian (EBG)

(Neill, 2006)

H_0: $q_i = 1$ everywhere
(counts = expected)

$H_1(S)$: $q_i = q_{in}$ in S and $q_i = 1$ outside, for some $q_{in} > 1$.
(counts > expected in S)

$$F(S) = \exp\left(\frac{(C')^2}{2B'} + \frac{B'}{2} - C'\right)$$

(if $C' > B'$)

Population-based Gaussian (PBG)

(Neill, 2006)

H_0: $q_i = q_{all}$ everywhere
(inside = outside)

$H_1(S)$: $q_i = q_{in}$ in S and $q_i = q_{out}$ outside, for some $q_{in} > q_{out}$.
(inside > outside)

$$F(S) = \exp\left(\frac{(C'_{in})^2}{2B'_{in}} + \frac{(C'_{out})^2}{2B'_{out}} - \frac{(C'_{all})^2}{2B'_{all}}\right)$$

(if $C'_{in} / B'_{in} > C'_{out} / B'_{out}$)

Which regions to search?

- Typical approach: each search region S is a subregion of the search space.
  - Choose some region shape (e.g. circles, rectangles) and consider all regions of that shape and varying size.
  - Low power for true events that do not correspond well to the chosen set of search regions (e.g. irregular shapes).
  - Heuristic search over irregularly-shaped regions does not guarantee finding the highest-scoring region.
- Alternate approach: each search region S represents a distinct subset of the locations.
  - Exponentially many possible regions: computationally infeasible for naïve search.
  - Some regions may be impossible or very unlikely; may need to constrain search to a subset of feasible regions.

The LTSS property

- In certain cases, we can search over the exponentially many subsets in linear time!
- Many commonly used scan statistics have the property of linear-time subset scanning:
    - Just sort the data records from highest priority to lowest priority according to some criterion…
    - … then search over groups consisting of the top-k highest priority records, for k = 1..N.

The highest scoring subset is guaranteed to be one of these!

The LTSS property

- Example: Poisson statistics (Kulldorff, EBP)
  - $F(S) = F(C, B)$, where $C = \Sigma c_i$ and $B = \Sigma b_i$ are the aggregate count and baseline of region S.
  - Sort locations s_i by the ratio of observed to expected count, c_i / b_i.
  - Given the ordering $s_{(1)} \ldots s_{(N)}$, we can **prove** that the top-scoring subset consists of the locations $s_{(1)} \ldots s_{(k)}$ for some k, $1 \leq k \leq N$.
  - This follows from the facts that F(S) is convex, increasing with C and decreasing with B.
- Also holds for Gaussian, nonparametric, …

How to use LTSS in practice?

- Simplest case: assume all subsets are equally likely (e.g. outbreak that does not cluster spatially)
  - LTSS gives highest-scoring subset by evaluating **N** subsets instead of **2^N** for naïve search.
  - Sample result: we can find the most anomalous subset of 97 western PA zip codes in **.03 sec** vs. **10^{24} years**.
- But what if we want to use spatial information to constrain our search over subsets?
  - Hard constraints: some subsets of locations are not allowed (e.g. non-contiguous or highly irregular regions).
  - Soft constraints: some subsets of locations are more likely than others. Maximize penalized likelihood ratio.
- In most cases, we cannot use LTSS directly to find the optimal subset subject to these constraints.

Fast localized scan

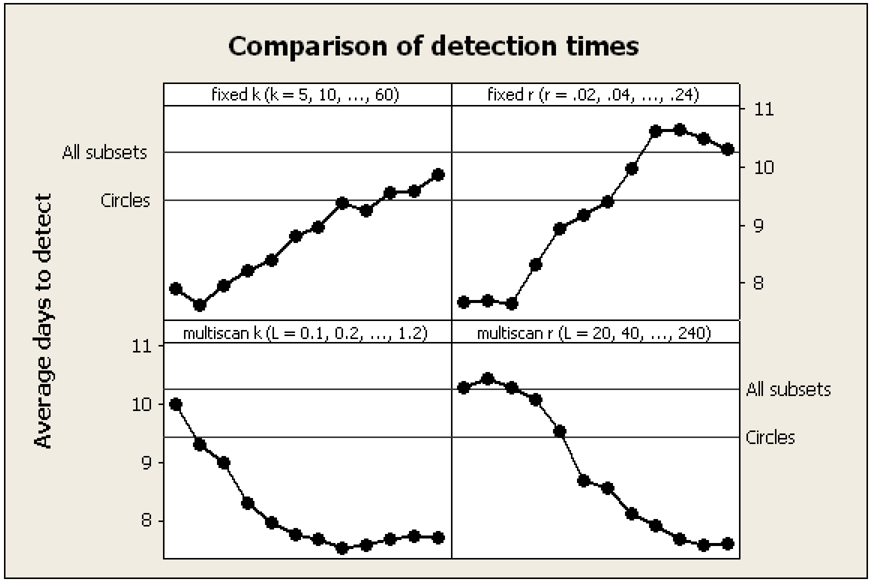
- Maximize the spatial scan statistic over regions consisting of a “center” location s_i and any subset of its k-nearest neighbors, for a fixed constant k (or fixed radius r).
- This is similar to Tango and Takahashi’s flexible scan statistic, but may find a disconnected region.
- Naïve search requires $O(N \cdot 2^k)$ time and is computationally infeasible for k > 25.
- For each center, we can search over all subsets of its k-nearest neighbors in O(k) time using LTSS, thus requiring a total time complexity of O(Nk) + O(N log N) for sorting the locations.

Fast localized multiscan

- Allow, but penalize, large neighborhoods.
- Perform separate “fast localized scans” for each neighborhood size k = 1…k_{max}.
- Choose the region which optimizes some function of likelihood ratio score F(S) and neighborhood size (or radius), e.g., F(S) – Lk.
- Computational complexity is $O(Nk_{max}^2 + N \log N)$.
- Timing results for 97 western PA zip codes:
  - Fast localized scan required up to 50 ms / day of data.
  - Fast multiscan required up to 1 sec / day of data.
  - Without LTSS, localized scan with k ≥ 40 would require over two years per day of data.

Evaluation of detection power

We compared the methods' average time to detect 2,000 simulated respiratory outbreaks (various sizes/shapes) injected into the real-world Emergency Department data from western PA.



All four proximity-constrained LTSS methods detected up to 1.8 days faster than circles, with less than half as many missed outbreaks.

Searching over all subsets, without proximity constraints, had poor detection power (0.8 days slower than circles).

Extensions of LTSS

LTSS can be easily extended to **space-time scans**: we must separately prioritize the locations and evaluate O(N) subsets for each temporal window size $W = 1 \ldots W_{max}$, with total run time $O(W_{max} N \log N)$.

We have recently developed a **fast graph scan**(1) which maximizes F(S) over all connected clusters, with or without proximity constraints. It can also be used for non-spatial graph data (e.g. contact tracing, social networks).

We can use LTSS to accelerate **multivariate** space-time scans(2):

➡ Burkom et al. (2005): add counts across the multiple streams, then apply the univariate statistic to the aggregate count.

Kulldorff et al. (2007): compute univariate LLR scores for each stream, then add scores across streams.

(1) S. Speakman & D.B. Neill, ISDS 2009.

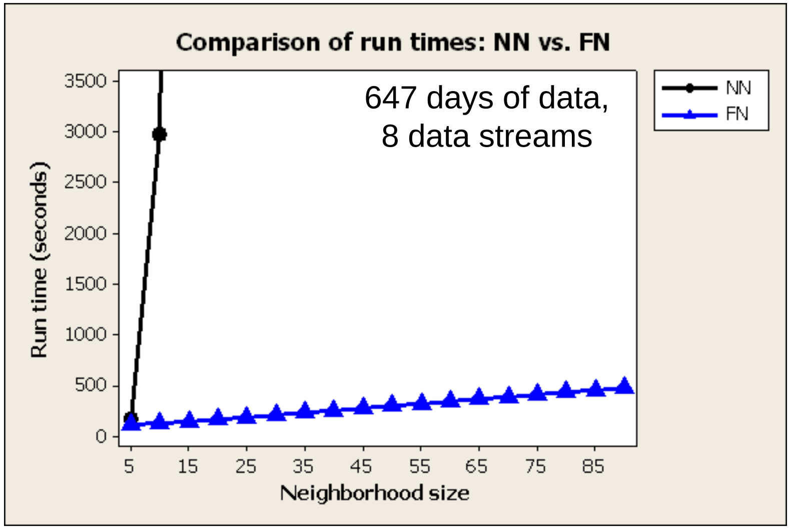
(2) D.B. Neill & E. McFowland III, in preparation.

Fast multivariate scans

Burkom's multivariate method loses detection power because high-count streams can "drown out" the signal: better is to search over all possible subsets of streams.

Option 1 (fast/naïve, or FN): for each of the 2^M subsets of streams, aggregate counts and apply LTSS to efficiently search over subsets of locations.

Guaranteed to find the highest scoring subset!



For a fixed number of streams, FN fast localized scan scales linearly (not exponentially) with neighborhood size.

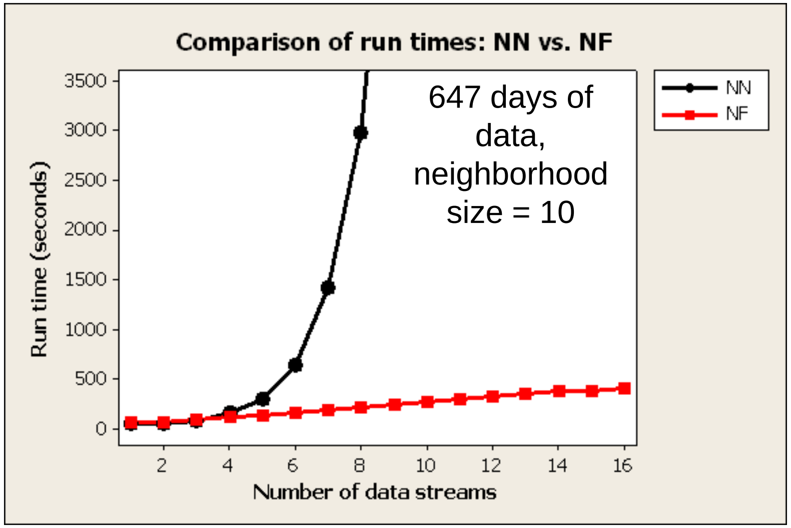
8 streams: <1 sec/day of data.

Fast multivariate scans

Burkom’s multivariate method loses detection power because high-count streams can “drown out” the signal: better is to search over all possible subsets of streams.

Option 2 (naïve/fast, or NF): exhaustively search over spatial regions. For each, perform efficient LTSS search over subsets of streams.

Guaranteed to find the highest scoring subset!



647 days of data, neighborhood size = 10

For a fixed neighborhood size k, NF fast localized scan scales linearly (not exponentially) with number of streams.

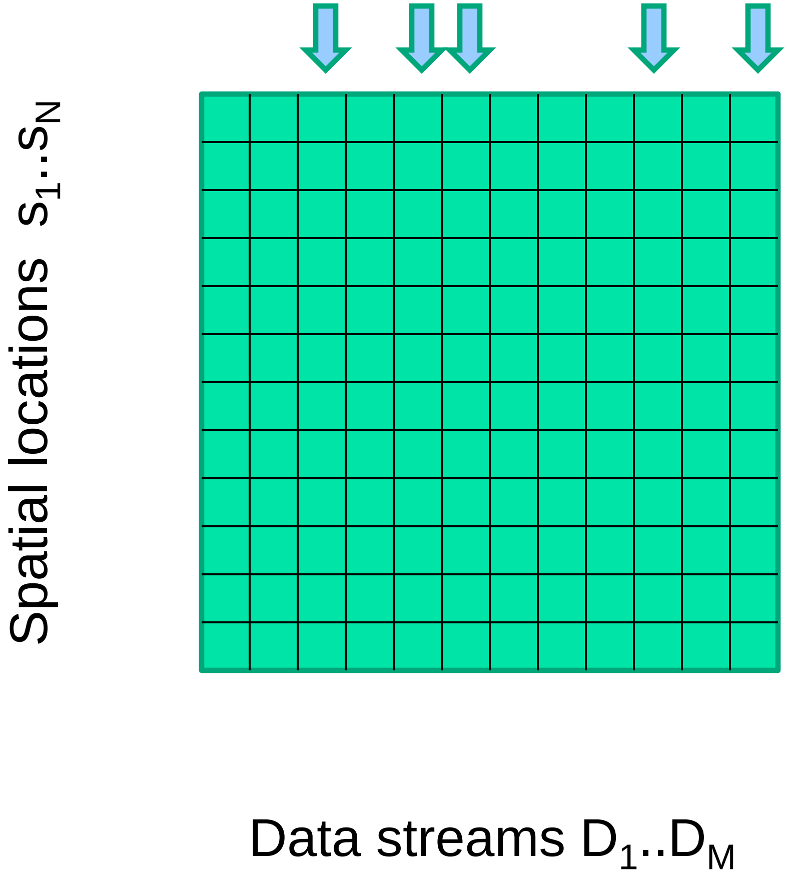
For k = 10: <1 sec/day of data

Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.



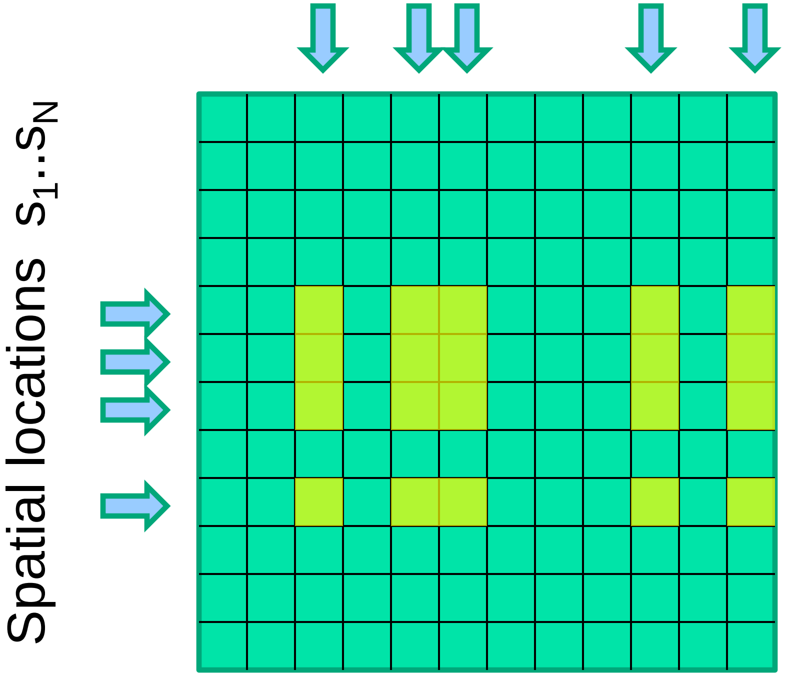
Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.



(Score = 7.5)

Data streams $D_1..D_M$

Fast multivariate scans

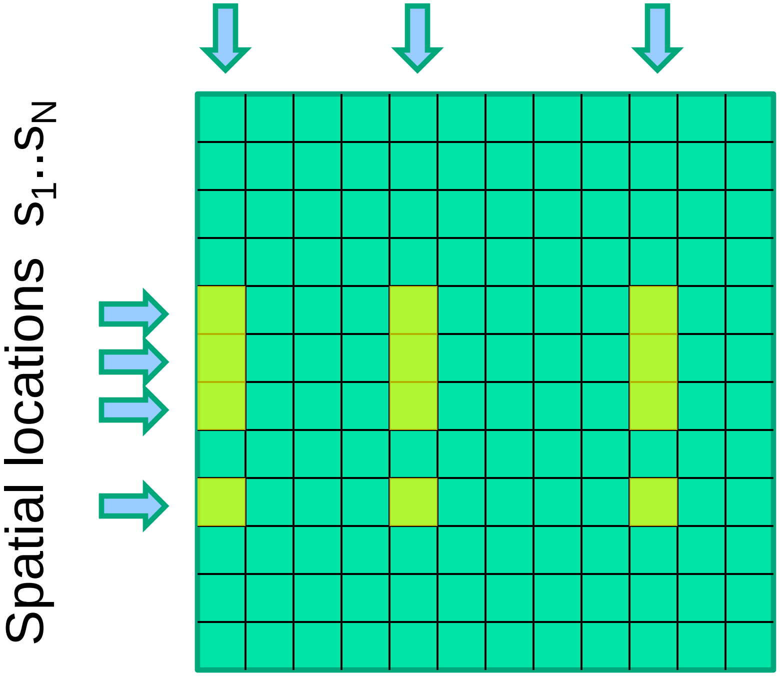
What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.



(Score = 8.1)

Data streams $D_1..D_M$

Fast multivariate scans

What if we have a large set of search regions and many data streams?

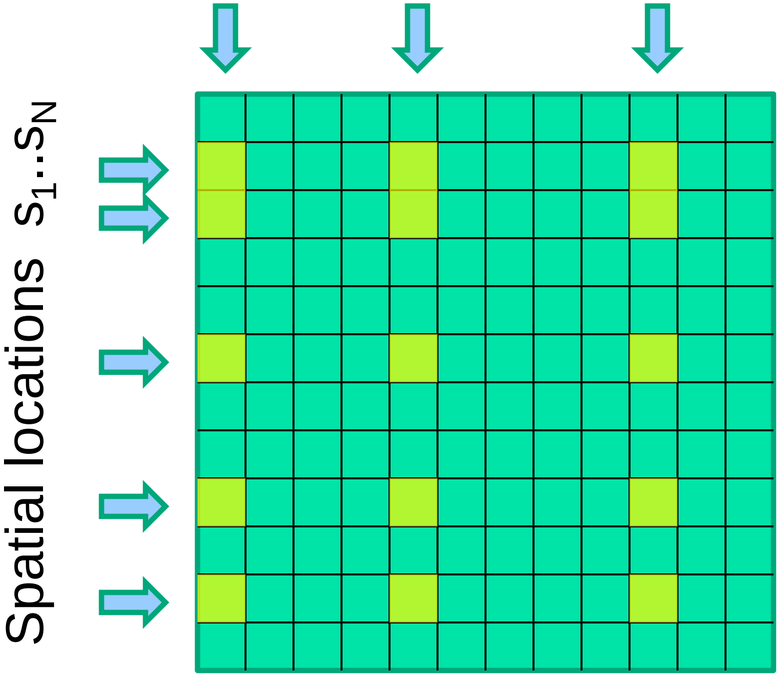
Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

4. Iterate steps 2-3 until convergence.



(Score = 9.0)

Data streams $D_1..D_M$

Fast multivariate scans

What if we have a large set of search regions and many data streams?

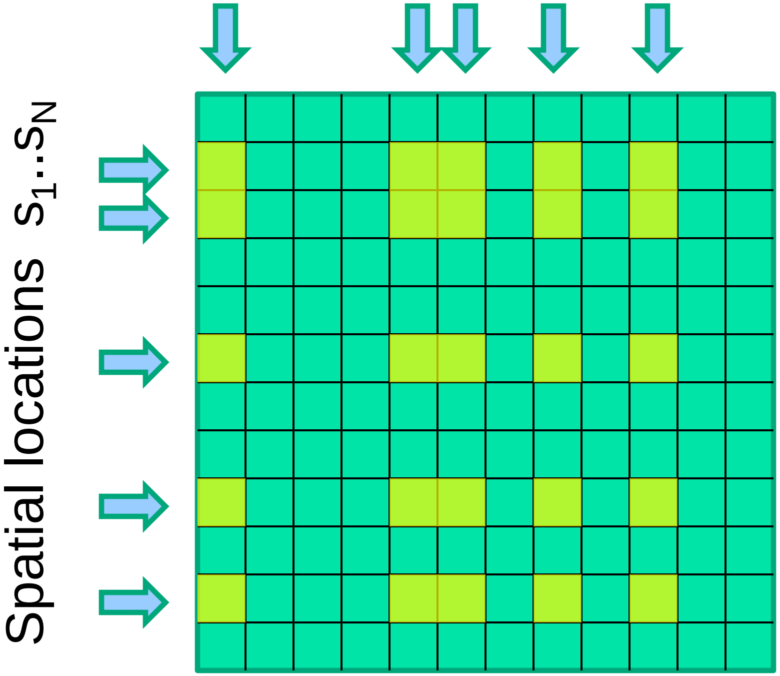
Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

4. Iterate steps 2-3 until convergence.



(Score = 9.3)

Data streams $D_1..D_M$

Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

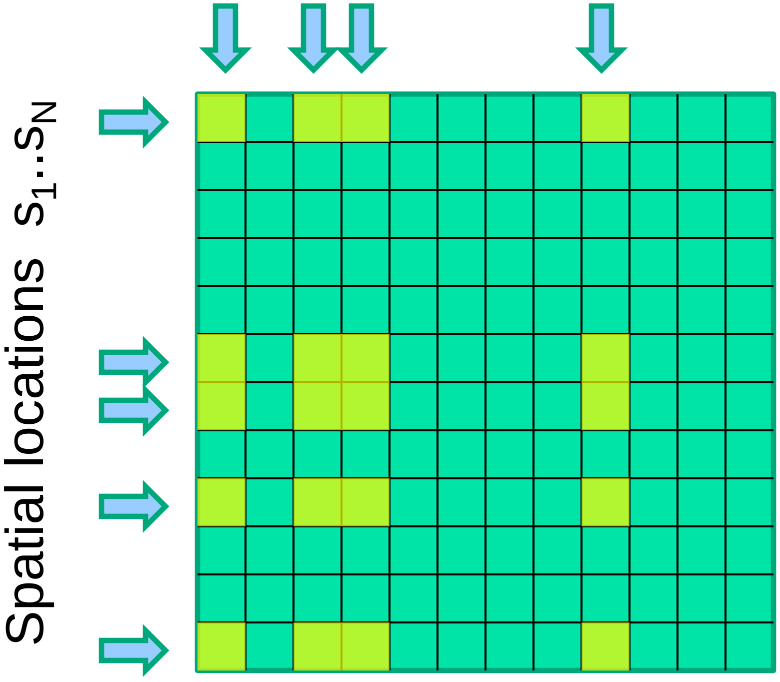
1. Start with a randomly chosen subset of streams.

2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

4. Iterate steps 2-3 until convergence.

5. Repeat steps 1-4 for 100 random restarts.



(Score = 11.0)

Data streams $D_1..D_M$

©2010 Carnegie Mellon University

Fast multivariate scans

What if we have a large set of search regions and many data streams?

Option 3 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

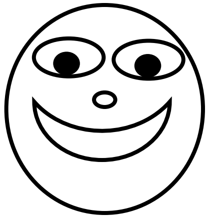
2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

3. Use LTSS to efficiently find the highest-scoring subset of streams for the given locations.

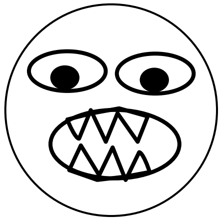
4. Iterate steps 2-3 until convergence.

5. Repeat steps 1-4 for 100 random restarts.

GOOD NEWS:
Run time is linear in number of locations & number of streams.



BAD NEWS:
Not guaranteed to find global maximum of the score function.



Fast multivariate scans

What if we have a large set of search regions and many data streams?

For neighborhood size = 15, number of streams = 16:

FF run time = 3.3 minutes for 647 days of data

42x speedup vs. NF; 88x speedup vs. FN

Accuracy: 65% exact, 90% within 5%, 97% within 10%.

For neighborhood size = 30, number of streams = 16:

FF run time = 4.0 minutes for 647 days of data

141x speedup vs. FN; NF infeasible

Accuracy: 57% exact, 89% within 5%, 98% within 10%.

Conclusions

Linear-time subset scanning is a powerful and useful tool that enables us to speed up a wide variety of spatial event detection methods.

The Poisson, Gaussian, and nonparametric spatial scan statistics all satisfy the LTSS property, as do many other possible statistics.

LTSS makes “all subsets” search, as well as proximity-constrained and graph-constrained scans, computationally feasible. The resulting methods significantly improve detection power and spatial accuracy.

Our recent extensions of LTSS to the multivariate and space-time scan statistics further increase the range of problems for which we can perform computationally efficient and fast event detection.