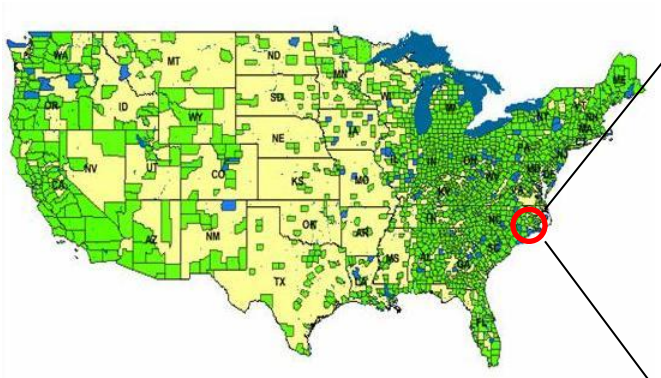
Spatial and Subset Scanning for Multivariate Health Surveillance

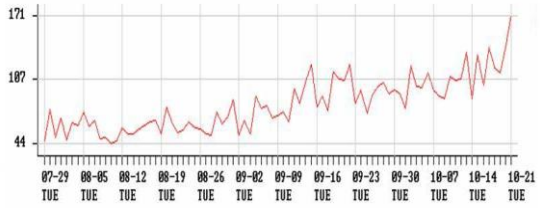
Daniel B. Neill, Ph.D.
Event and Pattern Detection Laboratory
Carnegie Mellon University
E-mail: neill@cs.cmu.edu

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

Spatial 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 location 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.)

Goals of detection task: **detect** any emerging disease outbreaks, **pinpoint** the affected spatial area, and **characterize** the type of event.

Informally, we want to know:

Is there anything happening?

If so, **what** and **where**?

Formally, we distinguish between:

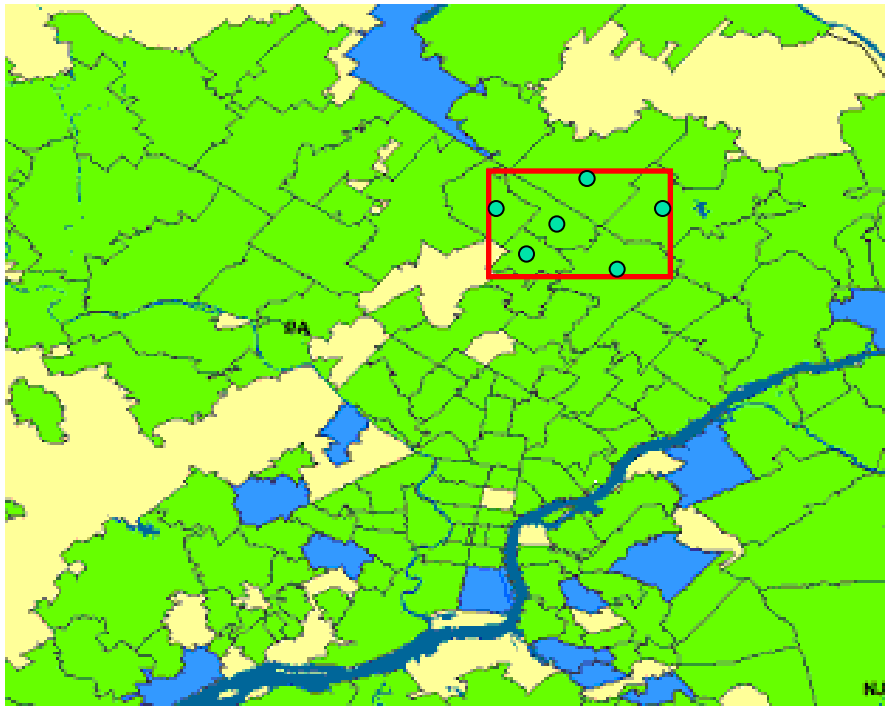
Null hypothesis $\mathbf{H_0}$ (no events)

Set of alternative hypotheses $\mathbf{H_1(S, E_k)}$ = event of type E_k in spatial region S.

(Spatial region = set of "nearby" locations, often constrain shape/size)

The spatial scan statistic

(Kulldorff, 1997)

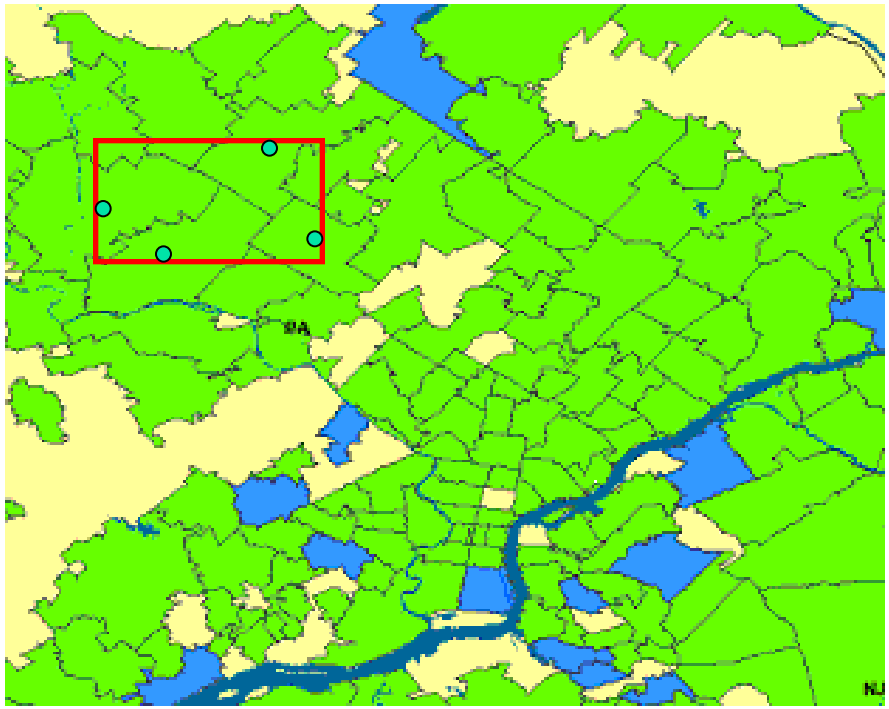


Rather than monitoring individual locations, we examine groups of locations.

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

The spatial scan statistic

(Kulldorff, 1997)

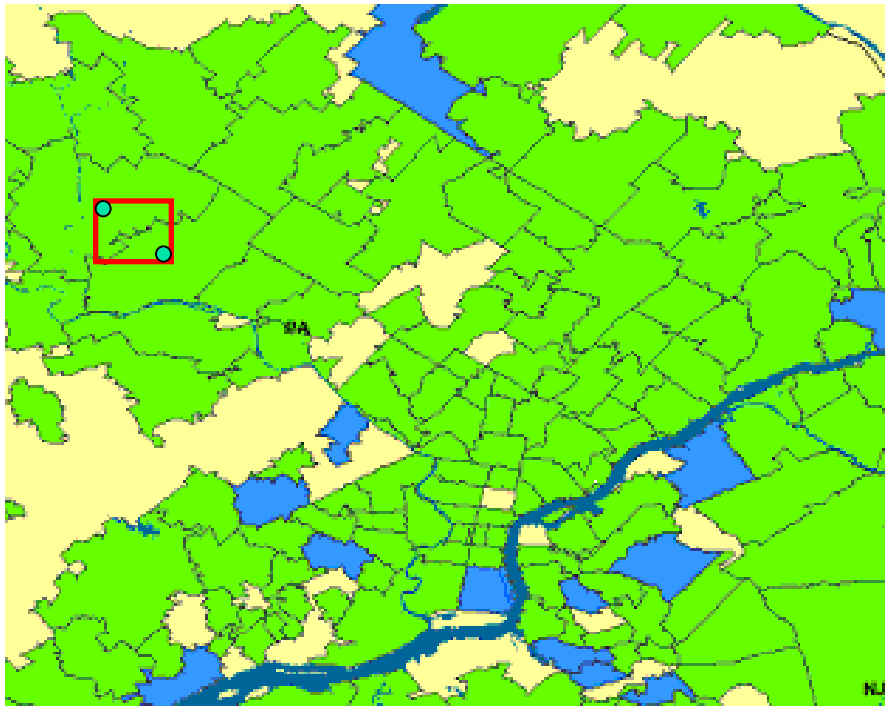


Rather than monitoring individual locations, we examine groups of locations.

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

The spatial scan statistic

(Kulldorff, 1997)

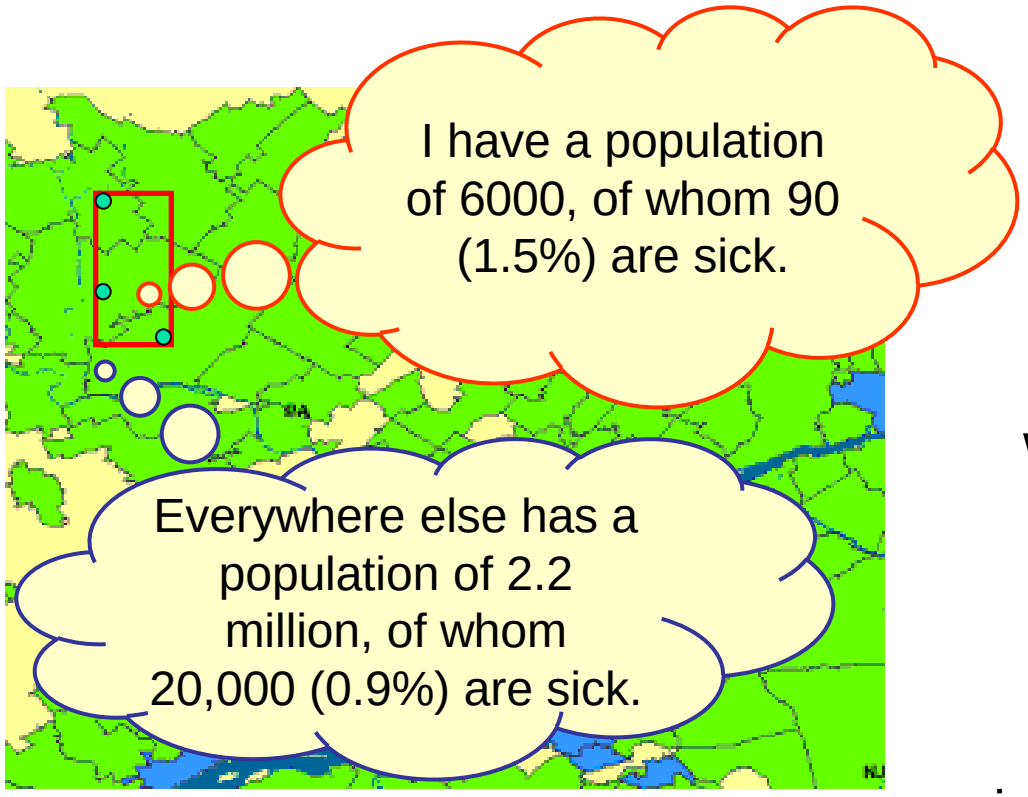


Rather than monitoring individual locations, we examine groups of locations.

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

The spatial scan statistic

(Kulldorff, 1997)



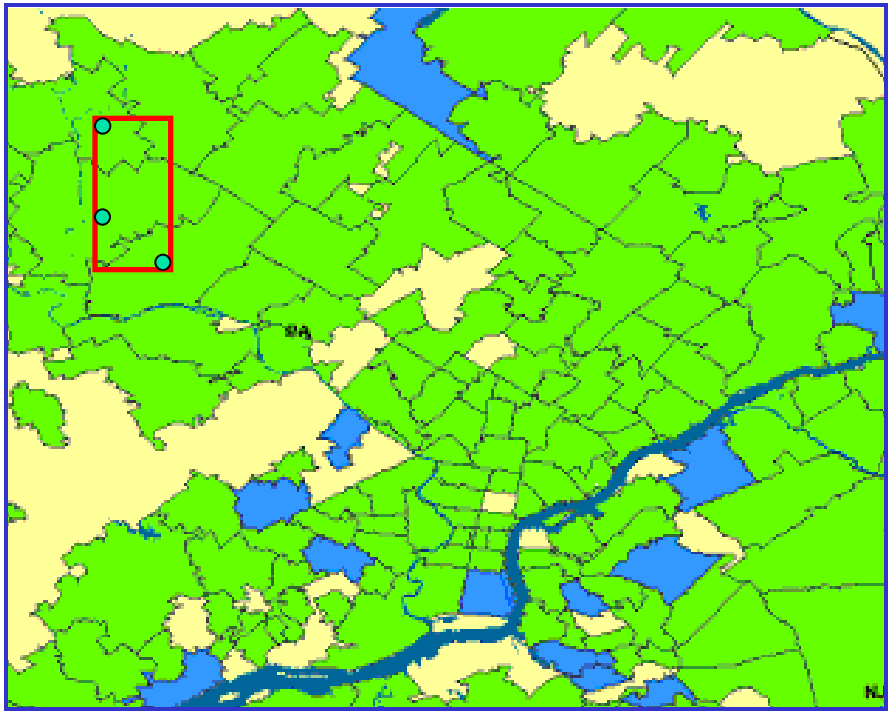
Rather than monitoring individual locations, we examine groups of locations.

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

Is there any position of the window such that the points inside form a significant cluster?

We compute a **score** for each spatial region, and then test whether the highest scoring regions are significant.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.

c_i = **count** for location s_i (e.g. number of disease cases)

b_i = **baseline** for location s_i (e.g. population at-risk, or expected count computed from historical data)

q = **risk** (expected ratio of count to baseline)

<u>Kulldorff's model</u>

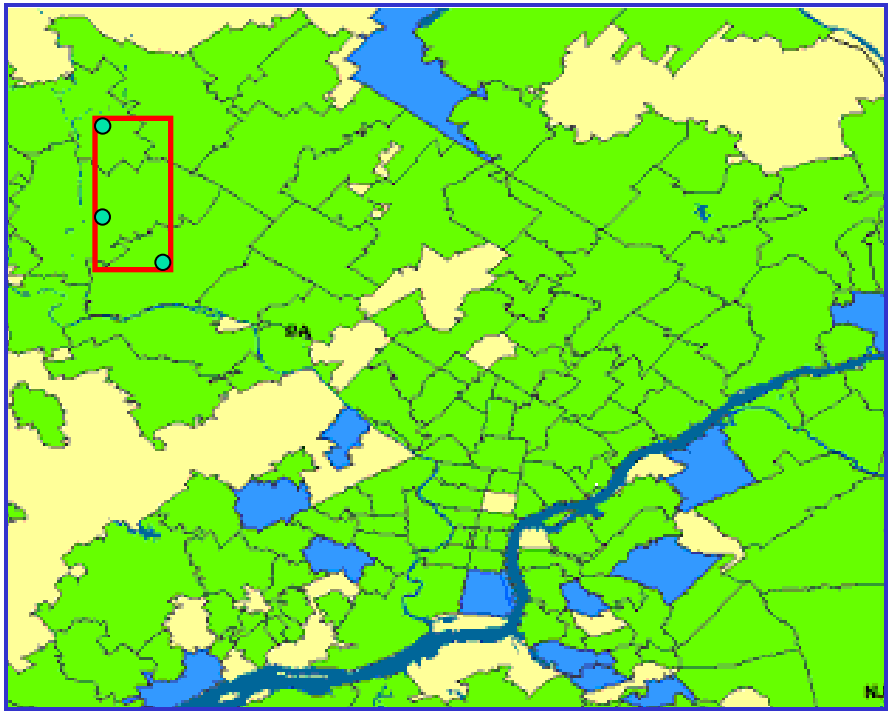
$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions

- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.



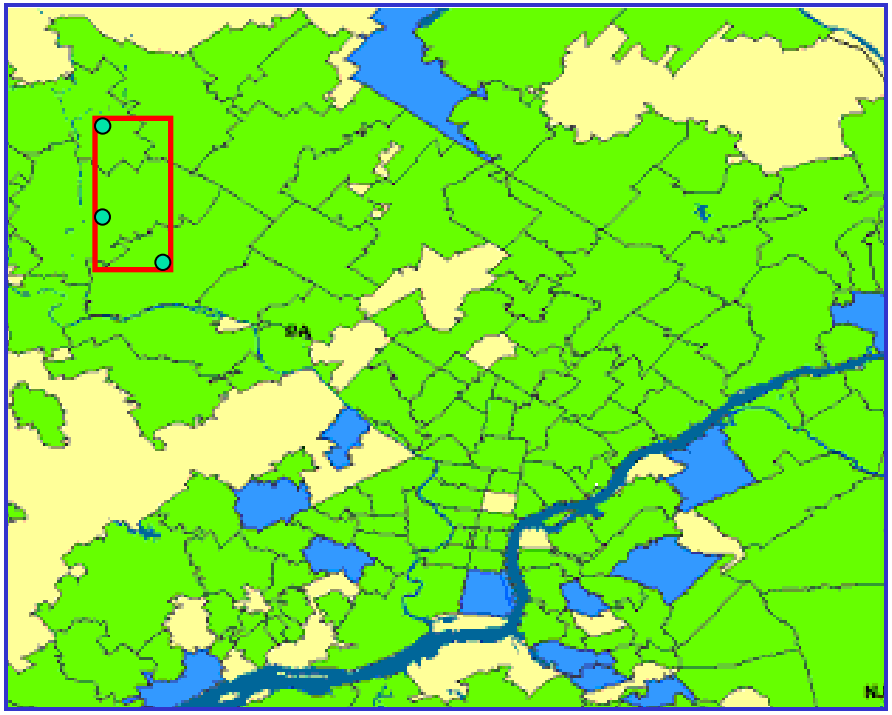
<u>Kulldorff's model</u>

$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.
- Derive a <u>score function</u>:
  - Likelihood ratio:

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

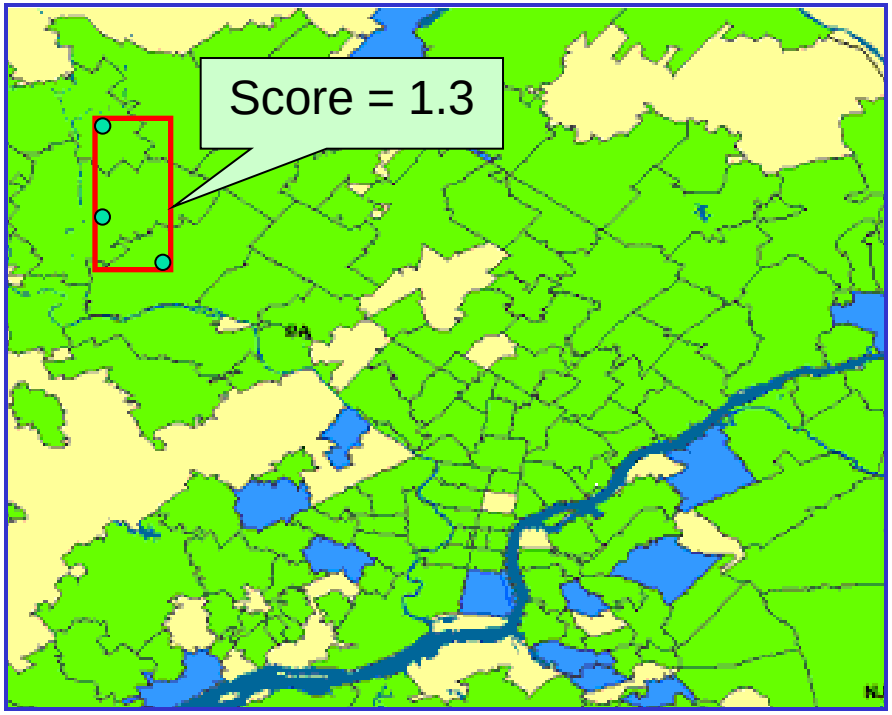
<u>Kulldorff's model</u>

$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.
- Derive a <u>score function</u>:
  - Likelihood ratio:

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

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

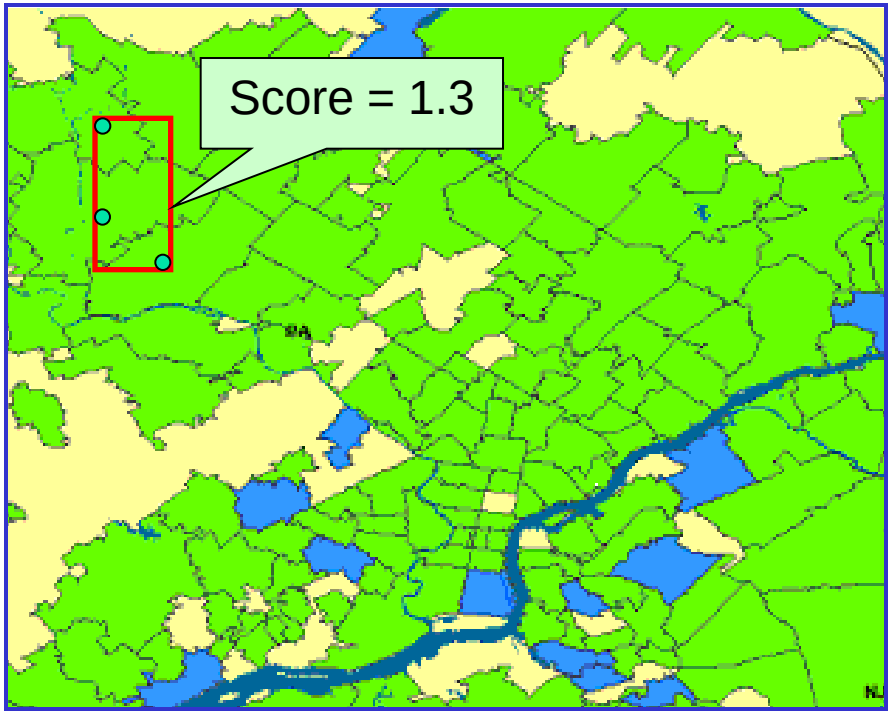
<u>Kulldorff's model</u>

$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.
- Derive a <u>score function</u>:
  - Likelihood ratio:

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

Total count and baseline of region S

Total count and baseline of search area

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

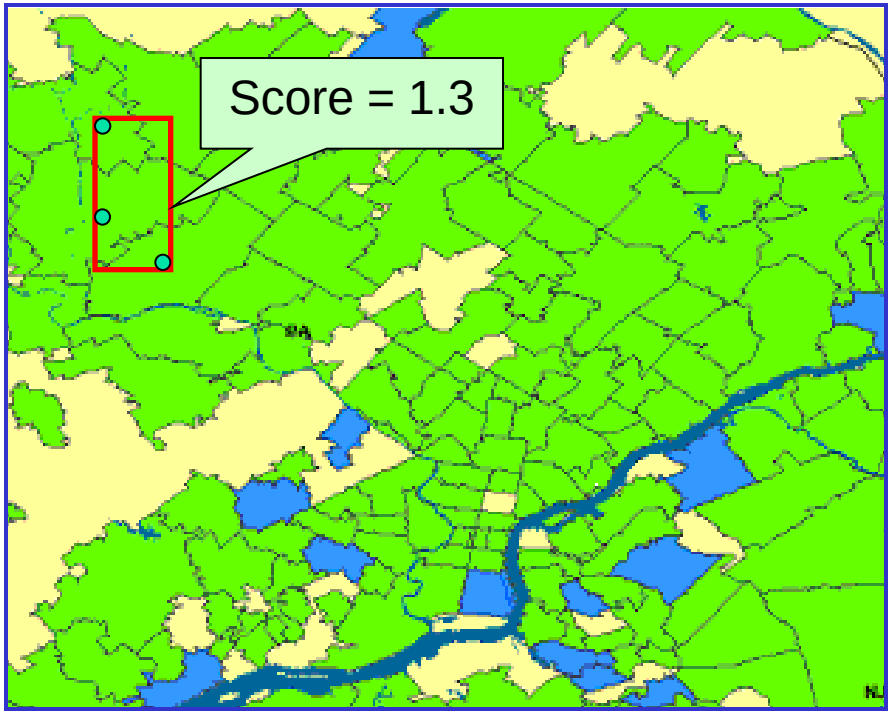
<u>Kulldorff's model</u>

$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.
- Derive a <u>score function</u>:
  - Likelihood ratio:

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

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

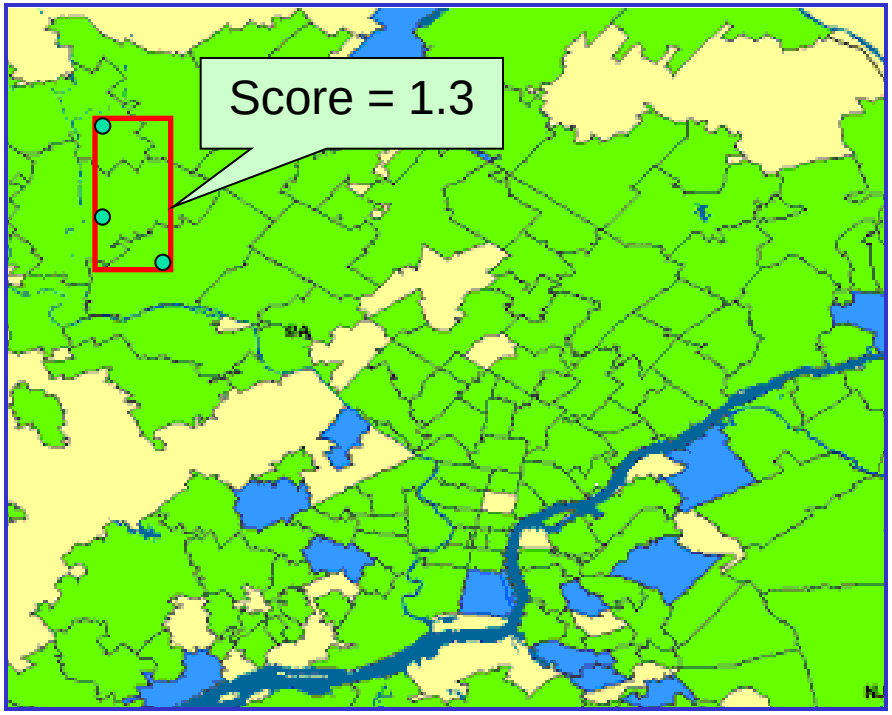
<u>Kulldorff's model</u>

$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.
- Derive a <u>score function</u>:
  - Likelihood ratio:

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

  - To find the most significant regions:

$$S^* = \arg\max_S F(S)$$

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

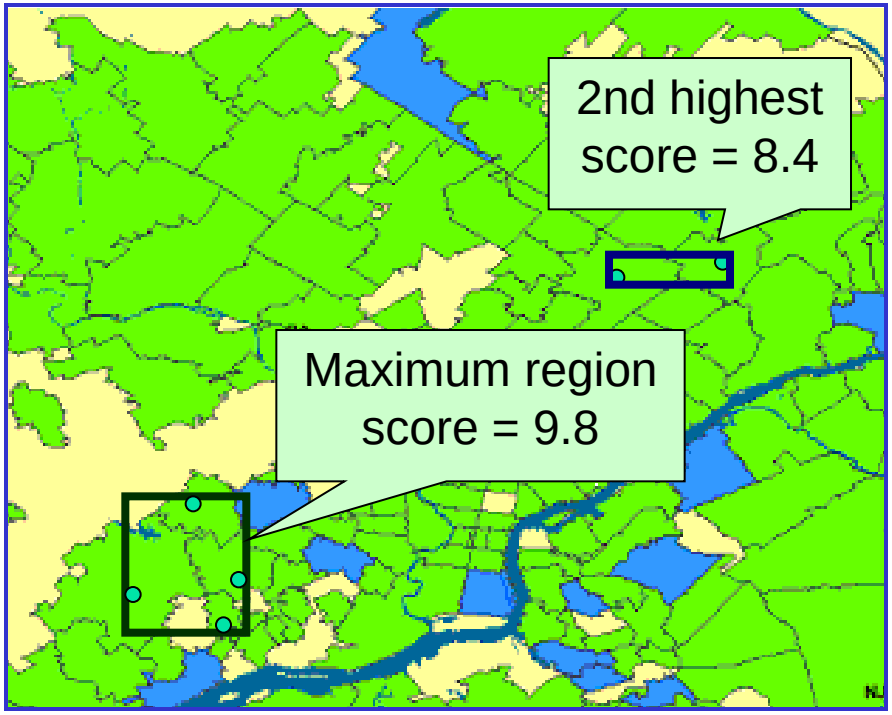
<u>Kulldorff's model</u>

$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

Finding the most significant regions



- Define <u>models</u>:
  - of the null hypothesis H_0: no events.
  - of the alternative hypotheses $H_1(S)$: event in region S.
- Derive a <u>score function</u>:
  - Likelihood ratio:

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

  - To find the most significant regions:

$$S^* = \arg\max_S F(S)$$

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

<u>Kulldorff's model</u>

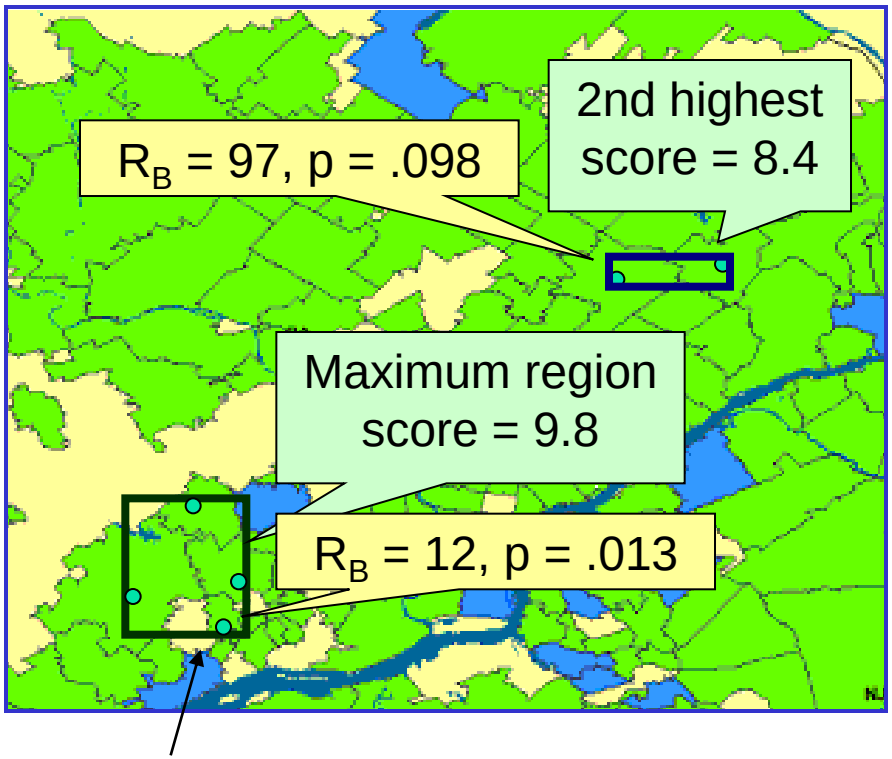
$c_i \sim \text{Poisson}(qb_i)$

H_0: $q = q_{all}$ everywhere

$H_1(S)$: $q = q_{in}$ inside S,
$q = q_{out}$ outside,
$q_{in} > q_{out}$.

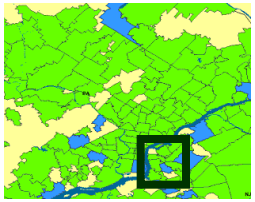
Which regions are significant?

- Randomly generate counts for R = 999 replica datasets under H_0 (i.e. assuming no events).
- Find maximum region score $F^* = \max_S F(S)$ of each replica.
- p-value of region S = $(R_B+1) / (R+1)$, where R_B = # of replicas with $F^* \geq F(S)$.
- All regions with p-values < α are significant at level α.



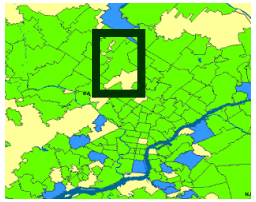
This region is significant at α = .05; no other regions are significant.

F* = 2.4



G_1

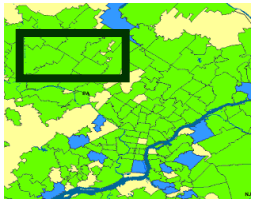
F* = 9.1



G_2

...

F* = 7.0



G_{999}

So what's missing?

- Typical spatial scan approaches may not be sufficiently flexible for our Data Fusion project.
- Need to integrate information from multiple, disparate streams of data.
- Searching over circles or rectangles is not sufficient:
  - In spatial settings, need to detect **irregular shapes**.
  - May have **connectivity constraints** instead. For monitoring nosocomial infections, consider flow of patients between hospitals or between rooms within a hospital.
  - May have **non-spatial** data. For example, we may wish to detect anomalous groups of "similar" patient records.
- Need to **scale up** to large numbers of records and streams.
- May want to distinguish between multiple, "known" event types as well as detecting previously unknown patterns.

Solution: subset scanning

- Rather than searching over spatial regions, search over **all subsets** of the data satisfying **constraints** on proximity, similarity, or connectivity.
- For example, we can consider all subsets of the "local neighborhood" of each record, and all subsets of the monitored streams. We find a group of **related** records with **anomalous** values for some subset of streams.
- Big problem: N records, M streams → 2^N x 2^M subsets to search!
- Our group has recently developed two fast and scalable approaches:

Fast subset scan: find the **best** subset of locations and data streams, subject to spatial proximity or graph connectivity constraints.

Fast subset sums: compute and visualize the posterior probability distribution over multiple locations and multiple event types.

Fast subset scan

- In certain cases, we can optimize F(S) over the exponentially many subsets of locations, while evaluating only O(N) rather than O(2^N) subsets.
- Many commonly used scan statistics have the property of linear-time subset scanning (LTSS):
  - Just sort the locations from highest to lowest priority according to some function…
  - … then search over groups consisting of the top-k highest priority locations, for k = 1..N.

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

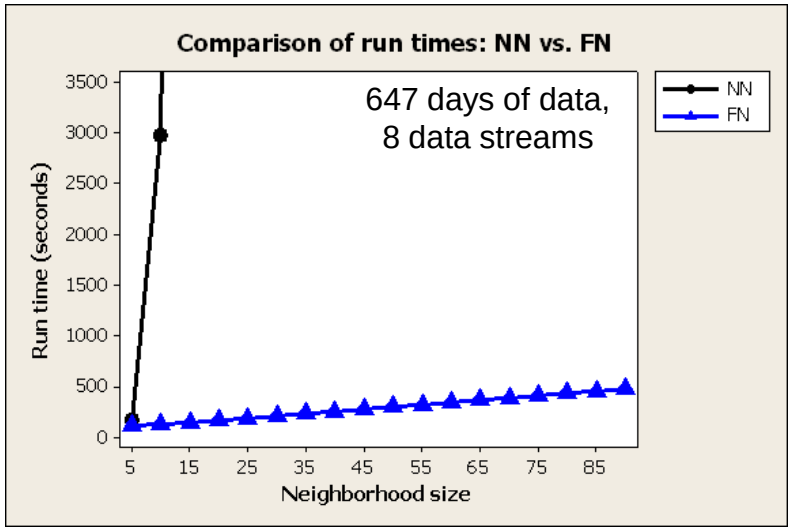
Sample result: we can find the **most anomalous** subset of Allegheny County zip codes in 0.03 sec vs. 10^{24} years.

Fast multivariate scans

Q: How can we efficiently search over all subsets of data streams and over all proximity-constrained subsets of locations?

A: We perform a separate, efficient search over the local neighborhood (e.g. k-nearest neighbors) of each data record.

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



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

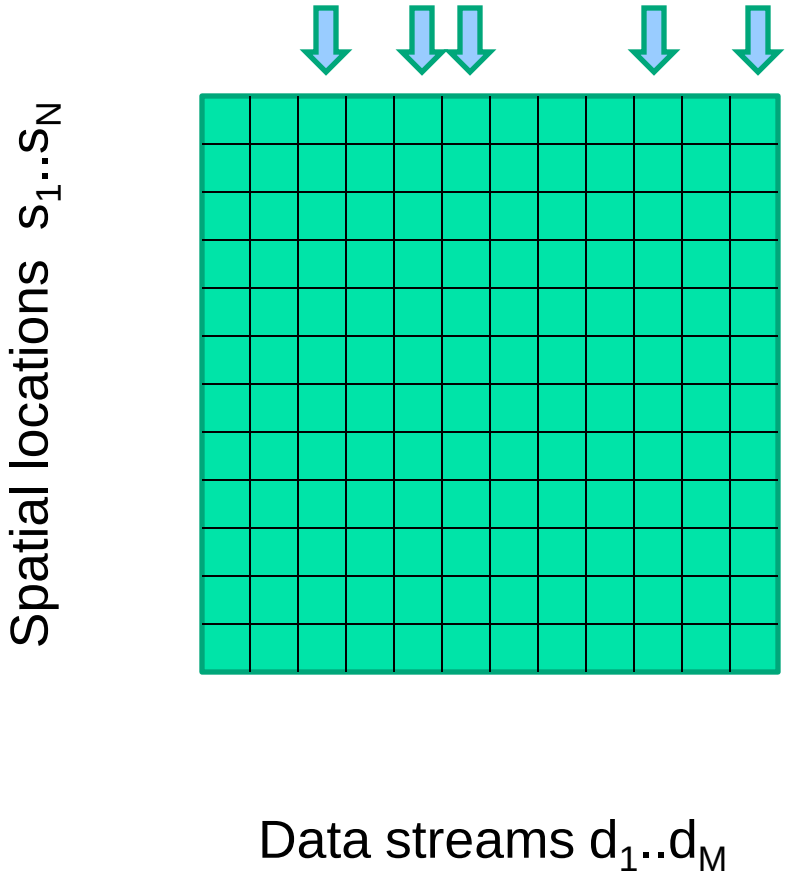
8 streams: <1 sec/day of data.

Fast multivariate scans

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

Option 2 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.



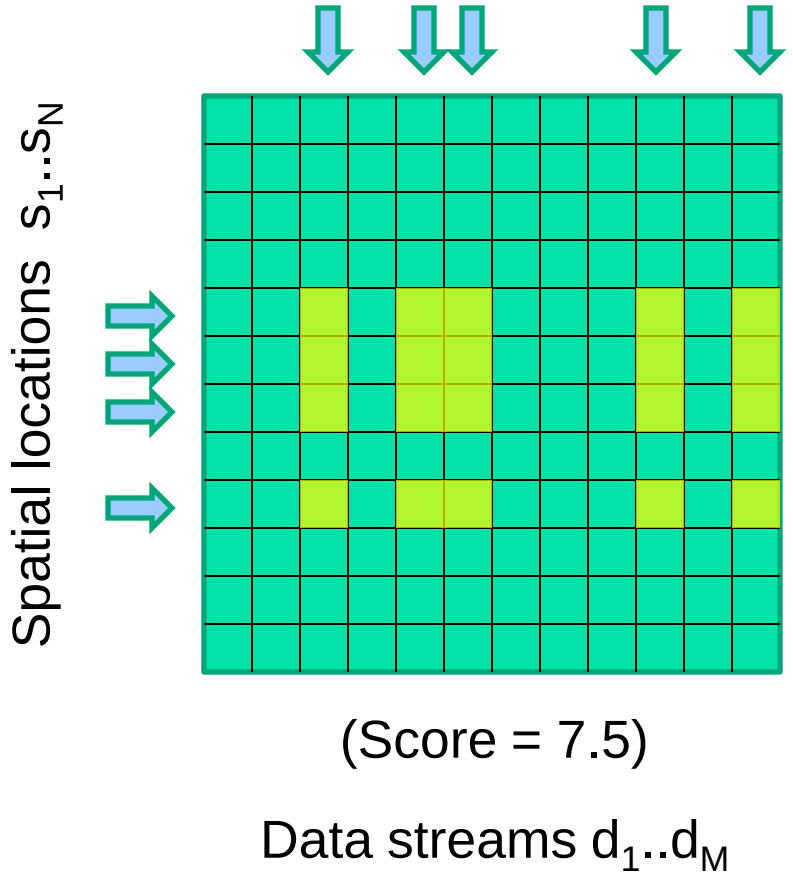
Fast multivariate scans

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

Option 2 (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.



Fast multivariate scans

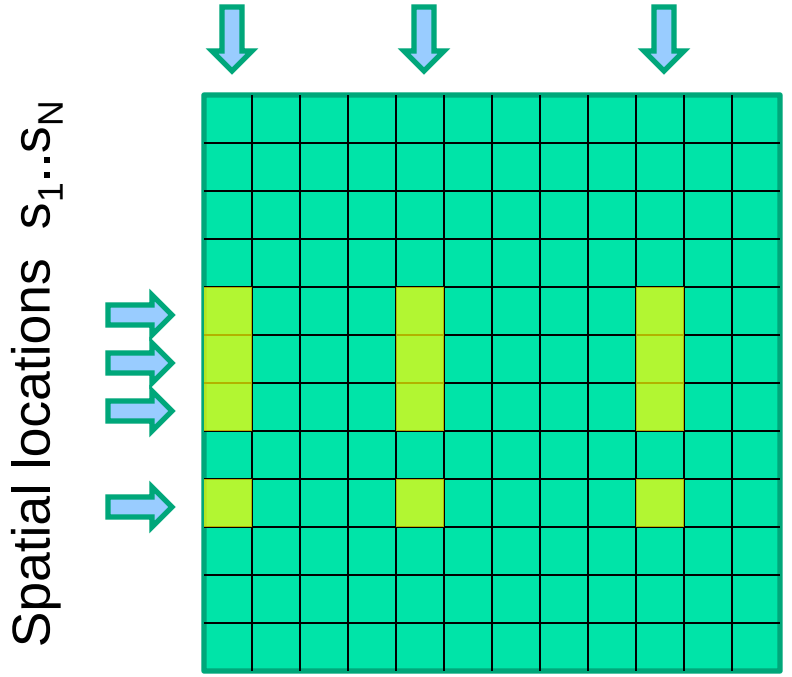
What if we have a large set of search regions and many data streams?

Option 2 (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.



Fast multivariate scans

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

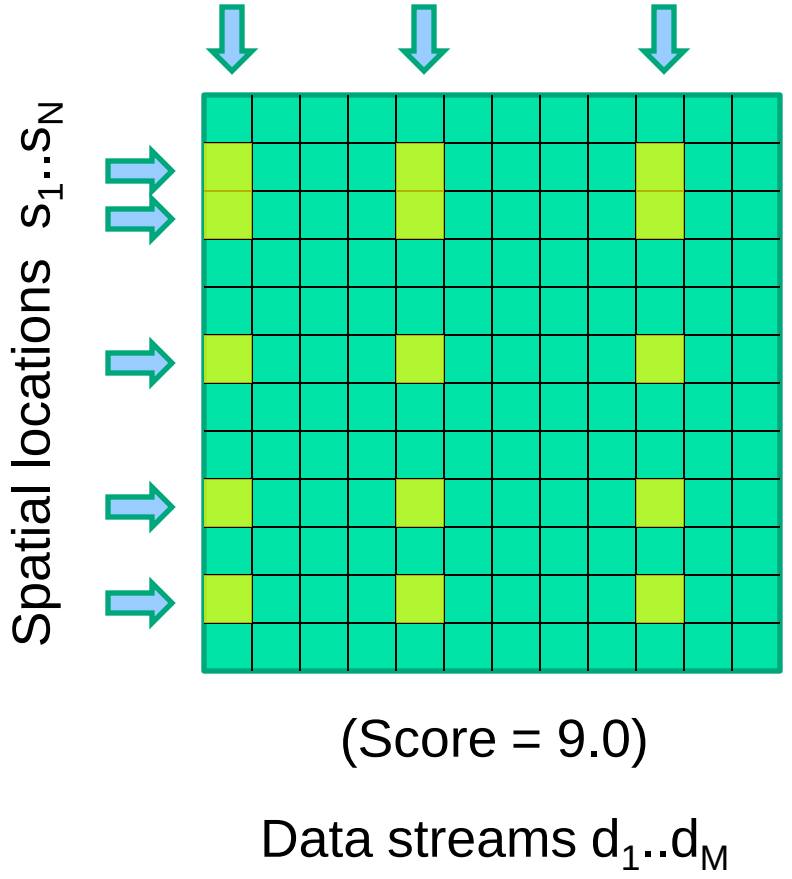
Option 2 (fast/fast, or FF):

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.



Fast multivariate scans

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

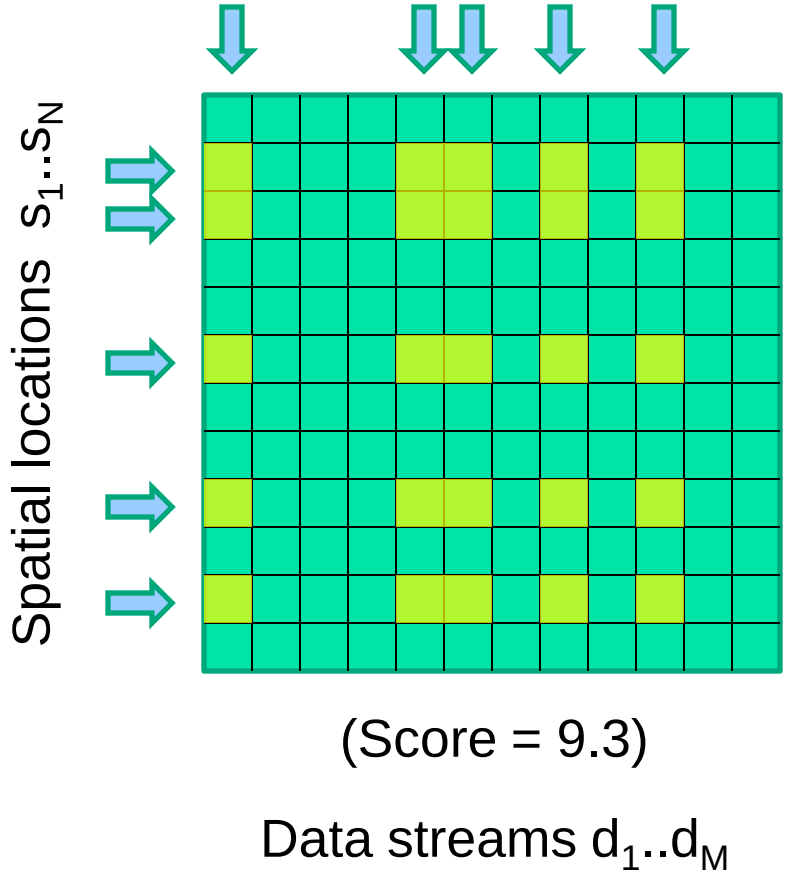
Option 2 (fast/fast, or FF):

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.



Fast multivariate scans

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

Option 2 (fast/fast, or FF):

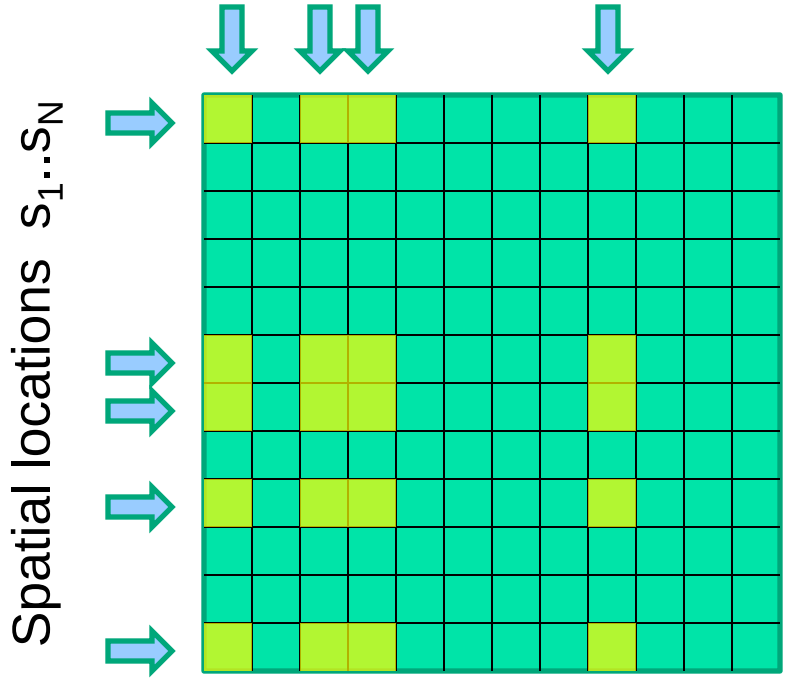
1. Start with a randomly chosen subset of streams.

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 50 random restarts.



Fast multivariate scans

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

Option 2 (fast/fast, or FF):

1. Start with a randomly chosen subset of streams.

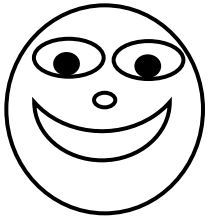
2. Use LTSS to efficiently find the highest-scoring subset of locations for the given streams.

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

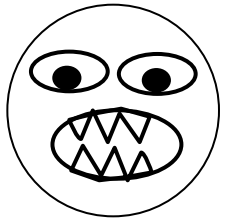
4. Iterate steps 2-3 until convergence.

5. Repeat steps 1-4 for 50 random restarts.

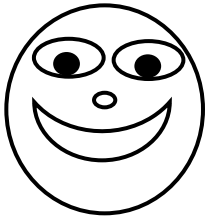
GOOD NEWS:
Run time is linear in number of locations & number of streams.



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



MORE GOOD NEWS:
200x faster than FN for 16 streams, and >98% approximation ratio.



Fast multivariate scans

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

Option 2 (fa... FF)

1. Start with ...

2. Us... the high... locations...

...find ...e ...nction...

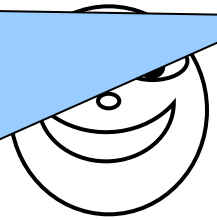
the ... streams...

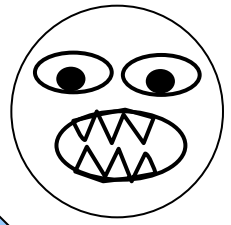
...ORE ...D NEWS: ...00x faster than FN for ...6 streams, and >98% approximation ratio.

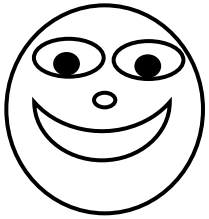
5. Repeat ... 1-4 for 50 rando... restarts.

What if, instead of finding the best subset, we want to obtain the **posterior probability** that each location has been affected, or to distinguish between multiple event types?

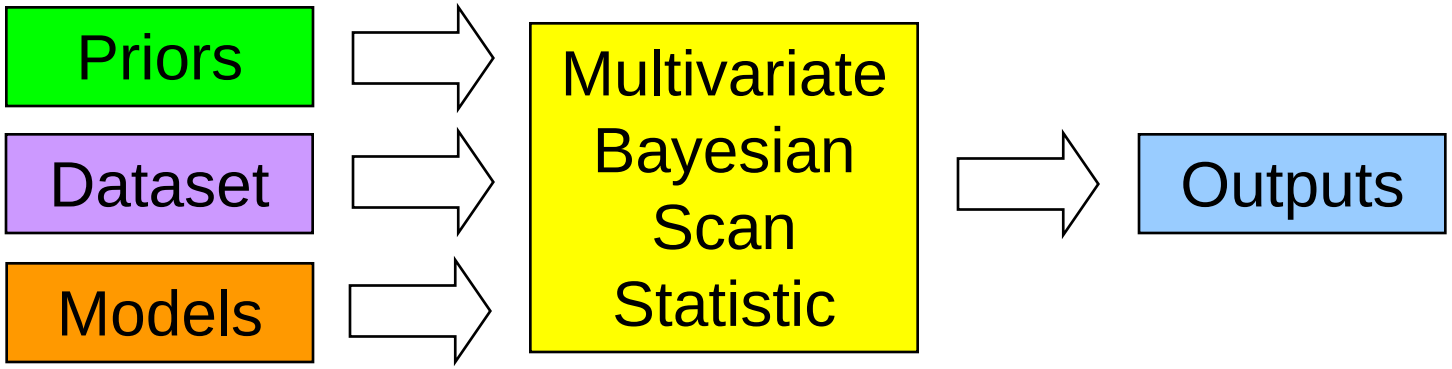
In this case, we should use "Fast Subset Sums" instead…







An extension of MBSS…

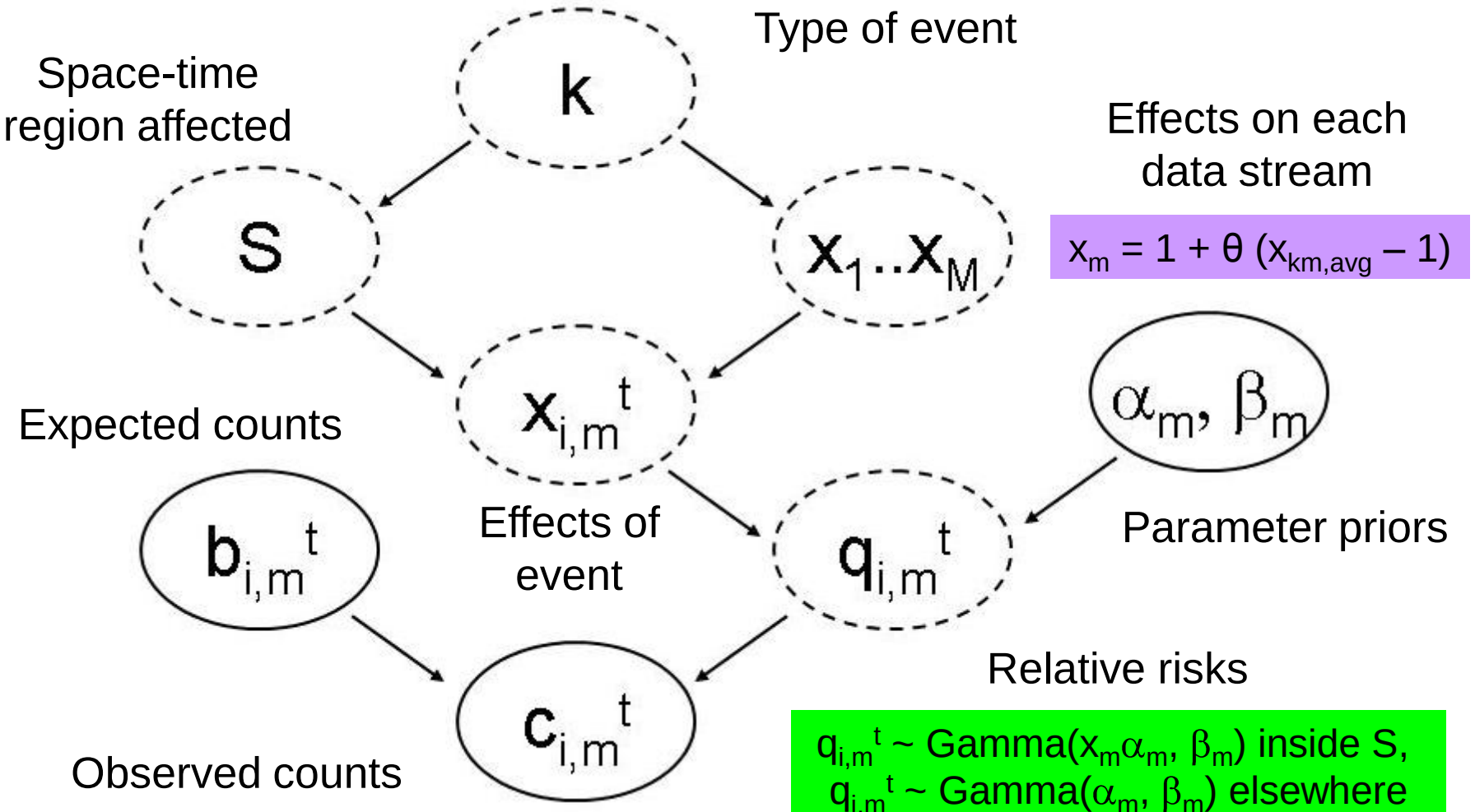


Given a set of event types E_k, a set of space-time regions S, and the multivariate dataset D, MBSS outputs the posterior probability $Pr(H_1(S, E_k) \mid D)$ of each type of event in each region, as well as the probability of no event, $Pr(H_0 \mid D)$.

We must provide the prior probability $Pr(H_1(S, E_k))$ of each event type E_k in each region S, as well as the prior probability of no event, $Pr(H_0)$.

MBSS uses Bayes' Theorem to combine the data likelihood given each hypothesis with the prior probability of that hypothesis: $Pr(H \mid D) = Pr(D \mid H)\ Pr(H) / Pr(D)$.

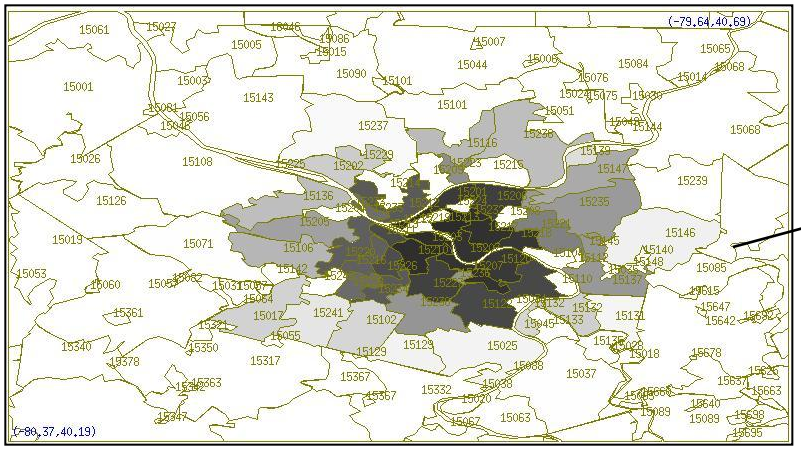
The Bayesian hierarchical model



Interpretation and visualization

MBSS gives the total posterior probability of each event type E_k, and the distribution of this probability over space-time regions S.

<u>Visualization</u>: $Pr(H_1(s_i, E_k)) = \sum Pr(H_1(S, E_k))$ for all regions S containing location s_i.



Posterior probability map

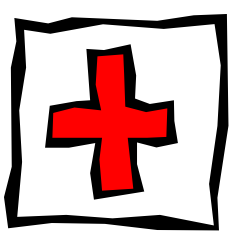
Total posterior probability of a respiratory outbreak in each Allegheny County zip code.

Darker shading = higher probability.

MBSS: advantages and limitations

MBSS can detect faster and more accurately by integrating multiple data streams.

MBSS can model and differentiate between multiple potential causes of an event.



MBSS assumes a uniform prior for circular regions and zero prior for non-circular regions, resulting in low power for **elongated** or **irregular** clusters.

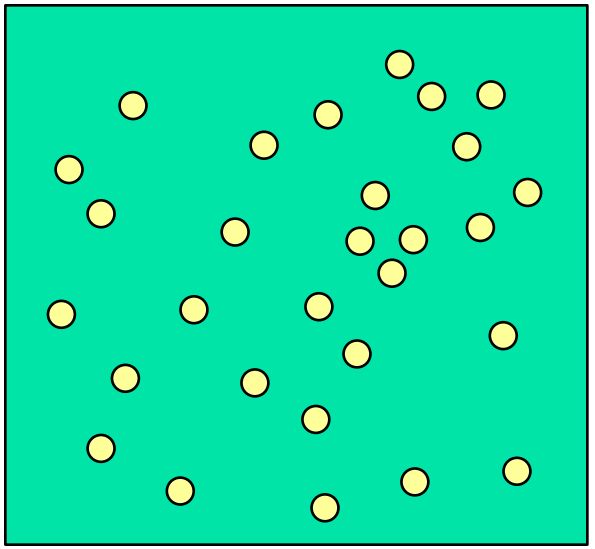
There are too many subsets of the data (2^N) to compute likelihoods for all of them!

How can we extend MBSS to **efficiently** detect irregular clusters?

The Fast Subset Sums method

We define a non-uniform prior $Pr(H_1(S, E_k))$ over all 2^N subsets of the data.

This prior has hierarchical structure:

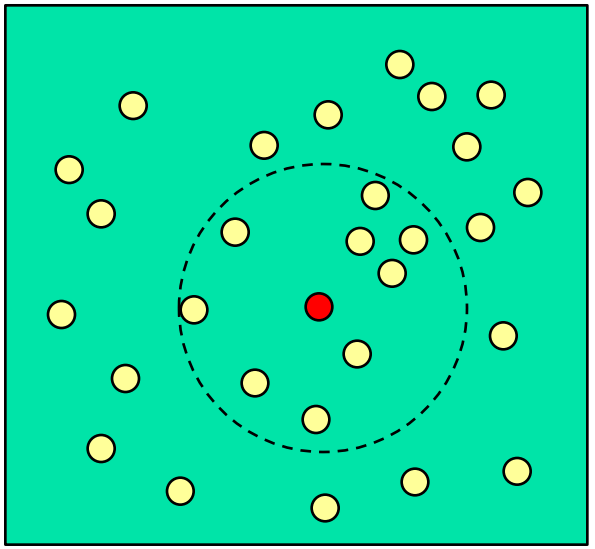


The Fast Subset Sums method

We define a non-uniform prior $Pr(H_1(S, E_k))$ over all 2^N subsets of the data.

This prior has hierarchical structure:

1. Choose **center location s_c** from $\{s_1 \ldots s_N\}$, given multinomial $Pr(s_i)$.
2. Choose **neighborhood size n** from $\{1 \ldots n_{max}\}$, given multinomial $Pr(n)$.

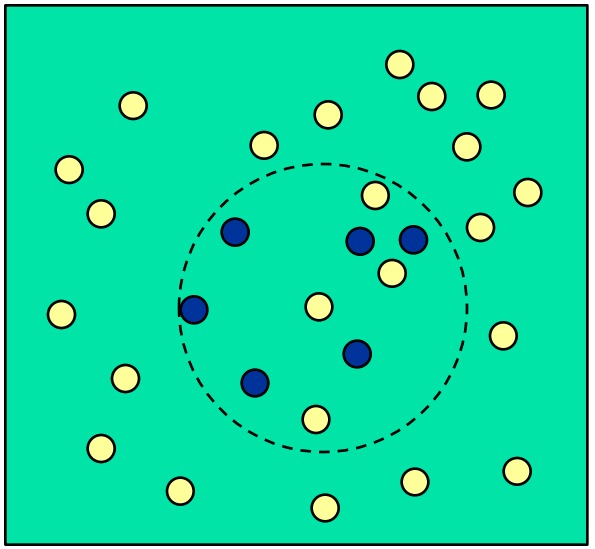


The Fast Subset Sums method

We define a non-uniform prior $Pr(H_1(S, E_k))$ over all 2^N subsets of the data.

This prior has hierarchical structure:

1. Choose **center location s_c** from $\{s_1 \ldots s_N\}$, given multinomial $Pr(s_i)$.
2. Choose **neighborhood size n** from $\{1 \ldots n_{max}\}$, given multinomial $Pr(n)$.
3. For each $s_i \in S_{cn}$, include s_i in S with probability p, for a fixed $0 < p \leq 1$.



This prior distribution has non-zero prior probabilities for any given subset S, but more compact clusters have larger priors.

Parameter p controls the sparsity of detected clusters.
Large p = compact clusters. Small p = dispersed clusters.

The Fast Subset Sums method

Naïve computation of posterior probabilities using this prior requires summing over an exponential number of regions, which is infeasible.

However, the total posterior probability of an outbreak, $Pr(H_1(E_k) \mid D)$, and the posterior probability map, $Pr(H_1(s_i, E_k) \mid D)$, can be calculated efficiently **without** computing the probability of each region S.

The main computational trick of FSS is just a bit of algebra: we can write the sum of 2^n products as a product of n sums.

More precisely, the **average likelihood ratio** of the 2^n subsets for a given center s_c and neighborhood size n can be found by multiplying the quantities $(p \times LR(s_i \mid E_k, \theta) + (1-p))$ for all locations s_i in S_{cn}.

FSS can compute the posterior probability map for ~100 locations in nine seconds, only a little slower than the original MBSS approach (searching over circles), and substantially improves detection power.

Conclusions

Our recent methods extend spatial scan to search over all **subsets** of the data, enabling more timely, flexible, and accurate event detection.

These methods enable us to integrate information from many data streams, scale to very large datasets, and incorporate relevant constraints on spatial proximity, graph connectivity, or similarity.

We address the computational challenge of searching over a huge number of subsets, efficiently finding either the best subset (“fast subset scan”) or the posterior probability map (“fast subset sums”).

The resulting methods solve optimization problems in seconds that would have previously required hundreds of millions of years, resulting in timely, accurate event detection in massive, multivariate data.