A software tool for creating simulated outbreaks to benchmark surveillance systems

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Published Version</td>
<td>doi:10.1186/1472-6947-5-22</td>
</tr>
<tr>
<td>Citable link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:4738539">http://nrs.harvard.edu/urn-3:HUL.InstRepos:4738539</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dashboard.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dashboard.current.terms-of-use#LAA</a></td>
</tr>
</tbody>
</table>
Software

A software tool for creating simulated outbreaks to benchmark surveillance systems
Christopher A Cassa*1,2,4, Karin Iancu2, Karen L Olson1,3 and Kenneth D Mandl1,3

Address: 1Children’s Hospital Informatics Program, Informatics Program – Mandl Group, 1 Autumn St, #721, Children’s Hospital Boston, Boston, MA 02115, USA, 2Clinical Decision Making Group, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA, 3Harvard Medical School, Boston, MA 02115, USA and 4Harvard/MIT Division of Health Sciences and Technology, Cambridge, MA 02139, USA

Email: Christopher A Cassa* - cassa@mit.edu; Karin Iancu - karini@alum.mit.edu; Karen L Olson - karen.olson@childrens.harvard.edu; Kenneth D Mandl - kenneth_mandl@harvard.edu

* Corresponding author

Abstract

Background: Evaluating surveillance systems for the early detection of bioterrorism is particularly challenging when systems are designed to detect events for which there are few or no historical examples. One approach to benchmarking outbreak detection performance is to create semi-synthetic datasets containing authentic baseline patient data (noise) and injected artificial patient clusters, as signal.

Methods: We describe a software tool, the AEGIS Cluster Creation Tool (AEGIS-CCT), that enables users to create simulated clusters with controlled feature sets, varying the desired cluster radius, density, distance, relative location from a reference point, and temporal epidemiological growth pattern. AEGIS-CCT does not require the use of an external geographical information system program for cluster creation. The cluster creation tool is an open source program, implemented in Java and is freely available under the Lesser GNU Public License at its Sourceforge website. Cluster data are written to files or can be appended to existing files so that the resulting file will include both existing baseline and artificially added cases. Multiple cluster file creation is an automated process in which multiple cluster files are created by varying a single parameter within a user-specified range. To evaluate the output of this software tool, sets of test clusters were created and graphically rendered.

Results: Based on user-specified parameters describing the location, properties, and temporal pattern of simulated clusters, AEGIS-CCT created clusters accurately and uniformly.

Conclusion: AEGIS-CCT enables the ready creation of datasets for benchmarking outbreak detection systems. It may be useful for automating the testing and validation of spatial and temporal cluster detection algorithms.
**Background**

The public health information infrastructure is yielding real-time access to health data, enabling new approaches to surveillance for infectious outbreaks. Prior to the laboratory confirmation or physician diagnosis of an infectious disease, ill persons may exhibit behavioral patterns, symptoms, signs, or laboratory findings that can be tracked through a variety of data sources. The process of monitoring these data is often referred to as syndromic surveillance [1-3].

Real time outbreak detection algorithms tend to focus on the temporal and spatial patterns of cases. Some detection routines, such as Cusum [4], look at just one type of pattern, while others, such as the Space-Time Scan Statistic [5] incorporate both. What these algorithms have in common is an underlying model of typical, or baseline patterns, and the goal of detection is to recognize perturbations from the baseline [1]. The outbreak detection performance of a surveillance system can be measured in terms of its ability to distill "signal" (a cluster of cases in time and/or space) from noisy baseline.

Benchmarking the performance of detection algorithms requires training and validation data. When real data are not available, simulated data are often used [6,7]. Simulated outbreaks must reflect the diversity of threats that a surveillance system is expected to encounter and detect, whether these outbreaks occur naturally or are man-made.

Our approach to validating detection algorithms is to use semi-synthetic data, that is, authentic baseline data injected with artificial signals [8]. These signals are defined by a controlled feature set of variable parameters such as the size, location, shape, and duration of simulated outbreaks. Here we describe a software tool, the AEGIS Cluster Creation Tool (AEGIS-CCT), that enables users to create simulated clusters with controlled feature sets, varying the desired cluster radius, density, distance, and relative location away from a central point. AEGIS-CCT does not require the use of an external geographic information system (GIS) program for cluster creation.

**Implementation**

**Functional specifications**

AEGIS-CCT can create single patient clusters as well as sets of patient clusters based on a simple geographical model. Cluster data points are outputted as comma separated variable (CSV) files, and can optionally be appended to an existing file, supplied by the user. The tool can also generate different sets of clusters that range in value over a single parameter to rigorously validate detection algorithms. To create individual clusters, the user can vary a number of relevant outbreak parameters (Table 1). AEGIS-CCT creates at least two output files each time it is executed, with file names specified by the user. One is a cluster data file that contains the artificial cluster data, and the other is a record file describing the session cluster parameters. The data file contains a cluster point identification number (assigned numerically from 0 to the number of points minus 1), the latitude and longitude of the cluster point, and the relative date of the cluster point. When generating a series of $n$ clusters, the program automatically generates $n$ files with appended identifiers, each as separate data files.

The AEGIS-CCT is a Java package including a geospatial engine and a user interface created using the Swing toolkit. The source for the entire package is provided under the Lesser GNU Public License [9] on a sourceforge.net development site [10]. Full details and updates to AEGIS-CCT can be obtained online [11].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster ID Number</td>
<td>User specified reference or identification number for each cluster</td>
</tr>
<tr>
<td>Number of Points in the cluster</td>
<td>Number of patients or points in the generated cluster.</td>
</tr>
<tr>
<td>&quot;Reference Point&quot; GIS Location</td>
<td>The latitude-longitude coordinates of a reference point, which could be a hospital or a primary care facility, for example.</td>
</tr>
<tr>
<td>Maximum cluster radius</td>
<td>The distance between the outermost point in the cluster and the center of the cluster.</td>
</tr>
<tr>
<td>&quot;Angle&quot; from the reference point</td>
<td>The angle of the cluster measured counter-clockwise from due east of the reference point as zero degrees, using unit circle convention.</td>
</tr>
<tr>
<td>Distance from the reference point</td>
<td>The distance between the center point of the cluster and the reference point.</td>
</tr>
<tr>
<td>Numbers of Days the Cluster should span</td>
<td>The number of days from when the first person shows symptoms to when the last person does.</td>
</tr>
<tr>
<td>Date Algorithm</td>
<td>This specifies which of the three models of temporal progression to use. Additional models can be incorporated into the software.</td>
</tr>
<tr>
<td>Description and output filenames</td>
<td>The user can specify where the cluster data and user-specified cluster description will be written.</td>
</tr>
</tbody>
</table>
Geocoding and precision of location

Programmatic methods were implemented to assign latitude-longitude coordinates to simulated cluster points, taking into account physical earth surface distances and not relying on external GIS software. Inside an AEGIS-CCT GIS class, there are three primary methods to handle these conversions. The first is a method to find the distance between two locations, which uses the specific latitude-longitude of the reference point to create a ratio of degrees per meter for north-south latitude and east-west longitude. Artificial data points are created 0.05 degrees to the north or east, and the corresponding physical distances (x, y) are calculated using the Haversine Formula, described below. The ratio is then computed, dividing the artificial data point distance by the calculated physical distance in meters on the Earth’s surface. A second method finds a point that is a specific physical distance, measured at a specified angle, from a reference point. The angle is measured from the Euclidian x-axis and increased in a counter-clockwise form. The output is a second GIS data point related to the reference point by the angle and distance specified by the user. The third method finds the number of degrees of latitude and longitude per unit of physical distance in each respective direction.

GIS precision

Placing patients on a map requires consideration of earth curvature and precise latitudes and longitudes. Spherical equations break down significantly at small distances, but the Haversine formula [12] provides computationally exact results in almost all circumstances. For this calculation, Earth has radius R, and the locations of two points in spherical coordinates (latitude and longitude) have names [lon1, lat1] and [lon2, lat2]. The Haversine Formula is calculated using the following code:

\[
dlon = \text{lon2} - \text{lon1};
\]

\[
dlat = \text{lat2} - \text{lat1};
\]

\[
a = (\sin(dlat/2))^2 + \cos(lat1) \cdot \cos(lat2) \cdot (\sin(dlon/2))^2;
\]

\[
c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a});
\]

\[
d = R \cdot c;
\]

This implementation was quality tested for accuracy using a series of latitude-longitude pairs from a sample dataset, measuring distances between two data points. Those results were identical with distances calculated by commercially-available GIS software.

Geotemporal progression

Outbreaks vary in their temporal progressions or epidemic curves. Three such progressions were implemented in AEGIS-CCT as date algorithms to model the ways in which a disease might manifest in a population over time: a random, a linear, and an exponential growth spread. Additional epidemiological date algorithms can be added by other users to the AEGIS-CCT by implementing a method to assign a specific temporal distribution within an array in Java.

For the random algorithm, a random number is generated that falls within the range of the number of days in the...
cluster, producing a random distribution. For the linear distribution, the day value is divided by the total day values and multiplied by the total number of points to determine the fraction of the total points to be injected each day. Scaling by a multiplier can alter the rate of linear growth. Similarly for the exponential distribution, the numerical value of $e^{(multiplier \times day\ number)}$ is divided by the sum of $e^{(multiplier \times day\ number)}$, for all day values. This ratio is then multiplied by the total number of points to determine the number of points that occur each day. Examples of the linear and exponential growth modified probability distribution estimations for date distribution are presented in Figures 1 and 2.

Results

Analysis of accuracy and uniformity of patient cluster data

Semi-synthetic datasets created by the cluster generator fall within a specified set of parameter-based boundaries. Cluster data points are created randomly within the domain defined by those parameters, so it is important to verify that the clusters are accurately created and are close to uniformly generated.

To measure the uniformity of generated clusters, 10 test clusters were created with 100 points in each cluster. The centroid of each set of cluster points was then calculated and compared to the specified center point of that cluster. In every case, the cluster centroid was within five percent of the specified cluster radius, in distance, from the specified center point. This result demonstrates that the datasets are uniform, within a small threshold, when they contain a sufficient number of points, as would be expected with a random distribution.

To measure the accuracy of the geocoding engine, 360 clusters were made, forming a circle, around a single center point, varying the angle evenly (one degree added per cluster,) and they each had a cluster center point that fell precisely along the circle defined by all points at the same radial distance from the original counterpoint. This same test was conducted at five randomly selected latitude-longitude locations and the same results were obtained.

Sample cluster parameters and output

Two sample scenarios are described below, and their parameters are listed in Table 2. The first example demonstrates the creation of a single artificial cluster and the second example demonstrates the creation of several clusters at various angles around a single reference point. The first cluster includes cases in a simulated outbreak spanning five days.

The single linear time-growth cluster was placed approximately 1600 m north of a center point at longitude -

---

Table 2: Example of single cluster parameters and multiple cluster parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Single Cluster</th>
<th>Type of Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster ID</td>
<td>100</td>
<td>101 [1-9]</td>
</tr>
<tr>
<td>Number of Points in Cluster</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Reference Point Latitude</td>
<td>42.35666</td>
<td></td>
</tr>
<tr>
<td>Reference Point Longitude</td>
<td>-71.09516</td>
<td></td>
</tr>
<tr>
<td>Cluster Radius</td>
<td>600 m</td>
<td></td>
</tr>
<tr>
<td>Angle from reference point</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Distance from reference point</td>
<td>1600 m</td>
<td></td>
</tr>
<tr>
<td>Number of Days</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Time-growth Pattern</td>
<td>Linear</td>
<td>Linear</td>
</tr>
<tr>
<td>Cluster Description</td>
<td>Linear time-growth cluster north of center point</td>
<td>Varied angle around center point and created 4 clusters.</td>
</tr>
<tr>
<td>Number of Clusters</td>
<td>N/A</td>
<td>4</td>
</tr>
<tr>
<td>Minimum Angle</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>Maximum Angle</td>
<td>N/A</td>
<td>270</td>
</tr>
</tbody>
</table>

Table 3: Sample output to a comma separated value file from AEGIS-CCT. Note: Values are point identification number, longitude, latitude and day number.

| 0,71.09600452536358,42.37455407329273,1 |
| 1,-71.1014972138236,42.36589456046806,1 |
| 2,-71.0954755413253,42.37389095443552,4 |
| 3,-71.0896377859539,42.37242100053542,2 |
| 4,-71.0928194633638,42.36955324904336,2 |
| 5,-71.09564524977307,42.371694560897,2 |
| 6,-71.09345472615571,42.37097950445030,4 |
| 7,-71.09983295495935,42.36968305959985,3 |
| 8,-71.09781606117451,42.37282937457113,3 |
| 9,-71.09685870199056,42.37540065852763,3 |
| 10,-71.09212141857045,42.37216701505921,3 |
... (to point with ClusterID 29)
71.09516 and latitude 42.35666. AEGIS-CCT outputted a CSV file and partial results are listed in Table 3. There are two points on the first day, four on the second day, six on the third day, linearly increasing to include thirty patient points by day 5.

The output CSV can be imported into a GIS analysis tool. A map made from AEGIS-CCT output using MapPoint 2002 (Microsoft Corporation, Redmond, WA) is presented in Figure 3. The temporal progression (linear-growth algorithm) is indicated by the shaded color of coordinate points for each simulated case as shown in the legend. The temporal growth pattern of a similar injected linear-growth cluster spanning 7 days (containing a total of 56 points) is graphed when appended to a low-volume week (120 visits, 07/15-21/2001) and a high-volume week (472 visits, 01/14-20/2001) in Figure 4.

In the second example, the angle around the center point was varied, creating a series of four clusters. The series cluster generator automatically created four files, and each file was imported into MapPoint, and charted in a different color, as shown in Figure 5.

The CCT can create new data files or it can append cluster data to an existing file. The artificial portion of a semi-synthetic dataset can then be automatically combined with baseline patient distributions by specifying a file that con-
tains the baseline data. In future versions, the CCT may provide an xml schema to which input baseline data files should conform.

**Limitations**

AEGIS-CCT does not yet have built-in procedures to generate more complicated time-series or spatial distributions. Extensibility was taken very seriously when creating AEGIS-CCT, and sufficient abstractions were made to allow for ease of adding additional models. The generator is also extensible in other ways so that additional parameters can be added if they are easily computable. Parameters can be added or deleted by updating the GUI and modifying the GenerateCluster method in the main geospatial class. Potential areas for expansion in future versions include log-linear and logarithmic time-series models as well as Gaussian spatial distributions. It will be necessary to determine what the most pertinent and physically realistic models of syndromic spread are. Once the most realistic scenarios are assessed and modeled, they can be programmatically implemented and incorporated into the cluster creation tool. As long as these distributions can be implemented using Java methods, it is possible to quickly add another temporal distribution to AEGIS-CCT.

**Conclusion**

Evaluation of surveillance systems for the early detection of outbreaks is particularly challenging [13] when the systems are designed to detect events for which there are a few or no historical examples. Some real-time surveillance systems are designed to provide early warning of a biological attack. Fortunately, few people have been infected with biological warfare agents, although there are notable exceptions. For example, residents of Sverdlovsk were exposed in 1979 during an accidental release of anthrax from a weapons plant [14] and there were eleven infections, resulting in five deaths in the Florida, New York and Washington DC mailed-anthrax attacks in 2001 [15]. In the absence of sufficient real outbreak data, measuring the detection performance of a system requires simulation. AEGIS-CCT enables the ready creation of datasets for benchmarking outbreak detection systems.

**Availability and requirements**

*Lists the following*

- Project name: AEGIS Cluster Creation Tool
- Project home page: http://sourceforge.net/projects/chipcluster/
- Operating system(s): Platform independent
- Programming language: Java
- Other requirements: Java 1.3.1 or higher
- License: e.g. GNU LGPL
- Any restrictions to use by non-academics: none

**List of abbreviations**

GIS – Geographical Information Systems
GUI – Graphical User Interface
CSV – Comma-Separated Values

**Competing interests**

The author(s) declare that they have no competing interests.

**Authors’ contributions**

CC implemented the GIS engine and dataset management, thoroughly tested of program, and was the primary author of the manuscript. KI helped implement the graphical user interface and the file read and write components. KO provided datasets and domain knowledge of specific issues relevant to the evaluation of detection performance. KM conceived of the study and participated in its design and coordination. All authors read and approved the final manuscript.

**Acknowledgements**

This work was supported by the National Institutes of Health through R01LM007970-01 from the National Library of Medicine, and by grant 2002-12-1 from the Alfred P. Sloan Foundation. This software was presented in part at the 2003 National Syndromic Surveillance Conference in New York, NY.

**References**

7. Goldenberg A, Shmueli G, Caruana RA, Fienberg SE: Early statistical detection of anthrax outbreaks by tracking over-the-


Pre-publication history
The pre-publication history for this paper can be accessed here:

http://www.biomedcentral.com/1472-6947/5/22/prepub