Disease surveillance systems

Baki Cakici

Licentiate thesis in Communication Systems
Stockholm, Sweden
2011

Disease surveillance systems
Baki Cakici
cakici@kth.se

Unit for Software and Computer Systems
School of Information and Communication Technology

Royal Institute of Technology (KTH)
Forum 120
SE-164 40 Kista

ISBN 978-91-7501-018-2
TRITA-ICT/ECS AVH 11:06
ISSN 1653-6363
ISRN KTH/ICT/ECS/AVH-11/06-SE

Abstract

Recent advances in information and communication technologies have made the development and operation of complex disease surveillance systems technically feasible, and many systems have been proposed to interpret diverse data sources for health-related signals. Implementing these systems for daily use and efficiently interpreting their output, however, remains a technical challenge.

This thesis presents a method for understanding disease surveillance systems structurally, examines four existing systems, and discusses the implications of developing such systems. The discussion is followed by two papers. The first paper describes the design of a national outbreak detection system for daily disease surveillance. It is currently in use at the Swedish Institute for Communicable Disease Control. The source code has been licenced under GNU v3 and is freely available. The second paper discusses methodological issues in computational epidemiology, and presents the lessons learned from a software development project in which a spatially explicit micro-meso-macro model for the entire Swedish population was built based on registry data.

Acknowledgements

I thank my supervisor, Magnus Boman, for his endless patience and support throughout my research. For their helpful comments on unreasonably short notice, I thank Björn Gambäck, Olof Görnerup, Anni Järvelin, Jussi Karlgren, and my co-supervisor Christian Schulte. I am grateful to Smittskyddsinstitutet, the Department of Analysis and Prevention, KTH Unit for Software and Computer Systems, and SICS Userware for hosting me during the past four years. My gratitude also goes to Marianne Hellmin for guiding me through the labyrinth of bureaucracy. I thank my mother, Iclal Cakici, for always pointing me towards the bright side of academia. Finally, I thank Hanna Sjögren for intellectual inspiration, for showing me ways to imagine otherwise, and for continually providing the possibility of optimism.

Contents

1 Overview **1**
- 1.1 Introduction . . . 1
- 1.2 Disposition . . . 3
- 1.3 State-of-the-art . . . 4
- 1.4 Constituents of disease surveillance systems . . . 5
  - 1.4.1 Collection . . . 5
  - 1.4.2 Analysis . . . 6
  - 1.4.3 Notification . . . 8
- 1.5 Implementations of disease surveillance systems . . . 9
  - 1.5.1 BioSense . . . 9
  - 1.5.2 RODS . . . 10
  - 1.5.3 BioStorm . . . 11
  - 1.5.4 HealthMap . . . 12
- 1.6 Discussion . . . 12
- 1.7 Advances on state-of-the-art . . . 14
- 1.8 Author's contributions . . . 15
- 1.9 A list of disease surveillance systems . . . 16

2 Papers **17**
- 2.1 CASE: a framework for computer supported outbreak detection 19
- 2.2 A workflow for software development within computational epidemiology . . . 27

References **45**

Chapter 1

Overview

1.1 Introduction

Surveillance is the act of monitoring and interpreting the activities of an object of interest. *Disease surveillance* is an epidemiological practise where the object of interest is defined to be a disease. Monitoring the disease host, or populations of potential disease hosts is implicit in the surveillance act; the disease cannot exist without the host. The potential and the immediate hosts are monitored for predefined signs, and the signs are interpreted in an attempt to prevent or minimise the spread of the disease. Disease surveillance is performed for both communicable diseases (influenza, chlamydia, salmonella, etc.) and non-communicable diseases (asthma, cancer, diabetes, etc.). The surveillance of the former is called *infectious disease surveillance*, and it most often involves analysing case reports or lab reports filed after doctors' visits. The lab verified results are often used as highly accurate indicators of the disease, but the delay between the onset of symptoms and the verification of the diagnosis may be several days to weeks depending on the disease, the diagnosis and the local infrastructure available to the practitioners. To reduce the delay, data originally collected for other purposes have been proposed as additional indicators to aid the understanding of infectious diseases. This approach is called *syndromic surveillance*, and its practitioners collect and analyse data from different data sources including pre-diagnostic case reports, number of hospital visits, over-the-counter drug sales and web search queries, among many other sources.

One of the most comprehensive and influential definitions of *syndromic surveillance* was given by the United States Centers for Disease Control and Prevention (CDC):

> Syndromic surveillance for early outbreak detection is an investigational approach where health department staff, assisted by automated data acquisition and generation of statistical signals, monitor disease indicators continually (real-time) or at least

> daily (near real-time) to detect outbreaks of diseases earlier and more completely than might otherwise be possible with traditional public health methods (e.g., by reportable disease surveillance and telephone consultation). The distinguishing characteristic of syndromic surveillance is the use of indicator data types. (Buehler et al. 2004, p.2)

Practitioners have criticised the usage of *syndromic surveillance* as imprecise and misleading, because many of the systems described by the term do not actually monitor syndromes, the association of signs and symptoms often observed together, but other non-health related data sources such as over-the-counter medication or ambulance dispatches (Mostashari 2003, Henning 2004). Despite its shortcomings, the term remains the most widely recognised among the alternatives. Other terms that describe similar or equivalent activities include early warning systems, prodrome surveillance, pre-diagnostic surveillance, outbreak detection systems, information system-based sentinel surveillance, biosurveillance systems, health indicator surveillance, nontraditional surveillance, and symptom-based surveillance. Some developers of syndromic surveillance systems have argued for broader terms, such as *biosurveillance*, to describe their work, in an attempt to unify outbreak detection and outbreak characterisation, claiming that epidemiologists may consider outbreak characterisation to be separate from public health surveillance (Wagner et al. 2006, p.3).

The unifying property of complex disease surveillance systems as information and communication systems is that they are designed to function without human intervention, performing statistical analyses at regular intervals to discover aberrant signals that match the parameters set by their operators. Recent advances in information and communication technology (ICT) have made the development and operation of such systems technically feasible, and many systems have been proposed to interpret multiple data sources, including those containing non-health related information, for disease surveillance. The introduction of these systems to the public health infrastructure has been accompanied by significant criticism regarding the diverting of resources from public health programs to the development of the systems (Sidel et al. 2002, Dowling & Lipton 2005), the challenges of investigating the alerts raised by the systems (Mostashari 2003), and the claims of rapid detection (Reingold 2003, Berger et al. 2006).

The motivation behind developing complex ICT systems for disease surveillance can be partially explained by the observation that epidemiologists tasked with monitoring communicable diseases are expected to maintain an awareness of multiple databases during their daily work. To provide the experts with a rapid overview of all available data, and to equip them with additional information to make decisions, development of ICT systems are proposed. Efficiently interpreting the combined output of these systems,

however, remains a technical challenge. In many cases, the populations represented in the data sources monitored by the systems differ significantly, preventing the application of traditional statistical methods to analyse the collected data. In theory, syndromic surveillance complements traditional disease surveillance in order to increase the sensitivity and specificity of outbreak detection and public health surveillance efforts. However, the syntactic and semantic diversity of the syndromic data sources complicates such efforts.

More importantly, the development of new methods of disease surveillance closely mirrors ongoing discussions in public health policy. The primary focus of syndromic surveillance on the unspecified and unexpected events challenges the traditional goals of public health. The goal of increasing the health of the populations through interventions against known events, improbable as they may be, is challenged by the mandate of preparedness, of defending against unknown or underspecified threats. To influence the direction of future research, discussing the implications of developing complex disease surveillance systems is of utmost importance today, while the field of syndromic surveillance is still in its infancy.

1.2 Disposition

The rest of chapter 1 describes the state-of-the-art in disease surveillance systems in more detail, presents a structural analysis of such systems, examines four existing implementations, and discusses the implications of developing and operating disease surveillance systems.

Chapter 2 includes two papers. The first, Cakici et al. (2010), describes the design of a national outbreak detection system inspired by syndromic surveillance systems. The system has been developed for daily communicable disease surveillance: the diagnoses monitored by the system are predefined, and the only data source used in detection is the communicable disease case database SmiNet (Rolfhamre et al. 2006). The system can perform different types of statistical analyses based on the users' preferences, and it regularly runs the requested analyses with the provided parameters. It is in use at the Swedish Institute for Communicable Disease Control.

The second paper, *A workflow for software development within computational epidemiology* (under review, Journal of Computational Science), discusses methodological issues in computational epidemiology, and presents the lessons learned from a software development project of more than 100 person months. The project is a spatially explicit micro-meso-macro model for the entire Swedish population built on registry data, thus far used for smallpox and for influenza-like illnesses. The list of lessons learned is intended for use by computational epidemiologists and policy makers, and the workflow incorporating these two roles is described in detail.

1.3 State-of-the-art

Considering the extensive history of public health literature, the development of complex information systems for disease surveillance is a recent addition. The first systems that proposed to monitor non-health related data sources for indicators of public health appeared in late 1990s (Heffernan et al. 2004*a*), and the development of larger systems began after 2001, most of them in the United States (Sosin & DeThomasis 2004). The last ten years have seen the development of a surprisingly large number of systems with diverse functionality. Out of this multitude, four systems were chosen in this work to highlight four corresponding directions taken by developers of disease surveillance systems:

- Integrating data collected from many institutions tasked with public health response to provide an overview of events concerning public health at the national level. BioSense, one of the largest syndromic systems ever deployed, accomplishes this by combining the data from diverse health facilities in 26 US states (Bradley et al. 2005).

- Understanding signals in many public health data sources in relation to each other, during the collection process, at the institute tasked with collection. RODS achieves this task by providing a self-contained system that can be deployed independently at multiple public health facilities (Tsui et al. 2003).

- Structuring the collected data and the methods of analysis in order to ease the difficulty of adding new sources or methods to existing systems. BioStorm provides ontologies that can classify analysis methods based on the goal of the analysis. It matches available data sources with suitable analysis methods (O'Connor et al. 2003).

- Increasing the visibility of results of disease surveillance analyses. The web-based HealthMap is accessible over the Internet without any additional authentication (Freifeld et al. 2008).

These systems are described further in section 1.5 below. The four properties of the examined systems are absent in traditional disease surveillance systems; they represent new contributions to the field, coinciding with the introduction of syndromic surveillance systems to public health practise. However, as the importance of ICT in disease surveillance increases, the boundary between syndromic and traditional non-syndromic surveillance blurs further. Systems designed for monitoring non-health related indicators grow to include diagnostic information, and systems for traditional disease surveillance begin to incorporate non-health related data sources.

Syndromic surveillance literature published in English is dominated by systems developed in and intended to be deployed in the United States, with

a few exceptions (Josseran et al. 2006, van den Wijngaard et al. 2011). More systems, developed in European states have been documented in the broader field of disease surveillance and outbreak detection (Hulth et al. 2010). Additionally, the European Commission has sponsored several large projects on Europe-wide systems for awareness and monitoring of pandemics, but the scope has been very wide and the software output has been modest; see, e.g., the INFTRANS (Transmission modelling and risk assessment for released or newly emergent infectious disease agents) project on the sixth framework programme, 2002–2006 (European Commission 2008).

1.4 Constituents of disease surveillance systems

Conceptually, disease surveillance systems may be partitioned into collection, analysis, and notification. The *collection* component contains lists of available data sources, collection strategies for data sources, instructions for formatting the collected data, and storage solutions. The *analysis* component stores a wide variety of computational methods used to extract significant signals from the collected data. The final component, *notification*, contains the procedures for communicating analysis results to interested parties. The results may be presented in many forms: numerical output from statistical analysis, incident plots displaying exceeded thresholds, maps coloured to indicate different levels of observed activity, or simply as messages advising the experts to check a data source for further information.

1.4.1 Collection

The set of accessible data sources is the most important factor in determining the capabilities of a disease surveillance system. Once again, following the growth of syndromic surveillance, a wide variety of sources have been proposed for monitoring. They may be divided into three groups based on when they become visible to the system relative to the patients' status: pre-clinical, clinical pre-diagnostic, and diagnostic (Buckeridge et al. 2002). An alternative method of categorising the data is to group according the type of patient behaviour that produces the data: information seeking after onset of symptoms; care seeking where the patient attempts to contact the healthcare provider or decides to purchase medication; and post-contact, when the patient becomes visible in traditional public health surveillance systems. Data sources most often used for syndromic surveillance, ordered by availability, from earliest to latest, are as follows (Berger et al. 2006, Babin et al. 2007):

- over-the-counter drug sales
- triage nurse line calls
- work and school absenteeism

- prescription drug sales
- emergency hotline calls
- emergency department visit chief complaints
- laboratory test orders
- ambulatory visit records
- veterinary health records
- hospital admissions and discharges
- laboratory test results
- case reports

In a survey of operational syndromic surveillance systems, Buehler et al. (2008) report that the 52 respondents monitor the following data sources: emergency department visits (84%), outpatient clinic visits (49%), over-the-counter medication sales (44%), calls to poison control centres (37%), and school absenteeism (35%). Another review by Chen et al. (2009) examines 56 systems and presents a comparable distribution of data source usage (p.37).

The availability of data sources depends on the local context of the project: jurisdiction of the organisation responsible for the system, diagnoses to be monitored, existing laws regulating data access, and technical concerns such as ensuring sustained connectivity to the data sources.

Recent research suggests that additional sources such as web search queries (Hulth et al. 2009, Ginsberg et al. 2009), and Twitter posts (Lampos et al. 2010) can also contain indicators for disease surveillance.

The timeliness of a data source is often inversely proportional to its reliability (Buckeridge et al. 2002). Sources with immediate availability such as Twitter posts or search queries often contain large amounts of false signals, and usually lack geographic specificity. In contrast, laboratory test results provide definitive diagnostic information, but they are not available early. An example between the two extremes is chief complaint records from emergency departments. These records are available on the same day as the visit, contain specific signs and symptoms as well as geographic information, but initially lack diagnoses (Travers et al. 2006).

1.4.2 Analysis

In traditional disease surveillance systems, the data forwarded by the collection component is associated with a diagnosis directly, and analysis begins. In syndromic surveillance systems, the data may contain signals for multiple diagnoses. Therefore, every data stream is assigned a *syndrome category* before it can be investigated for statistically significant signals. Syndrome categories are lists of signs and symptoms that indicate specific diseases; examples include *respiratory*, *gastrointestinal*, *influenza-like*, and *rash*. The assignment proceeds in two steps: first, the information relevant to the categorisation is extracted from the collected data, and second, the extracted

information is used to associate the data with the syndrome category. The extraction procedure is trivial for pre-formatted data sources such as over-the-counter drug sales (the data are already categorised by drug type), but may require complex methods for free-text data sources such as emergency department chief complaints. The extracted information is then associated with one or more syndrome categories, either by using a static mapping of data sources to diseases, or by an automated decision-making mechanism. An example of the former is CDC's syndrome categories in BioSense, and of the latter, BioStorm's ontology-driven assignments. Both systems are described in more detail in the next section. In existing syndromic surveillance systems, Bayesian, rule-based, and ontology-based classifiers have been used to assign syndrome categories (Chen et al. 2009, p.53).

After the categories are assigned, statistical analysis is used to detect significant signals. These signals may be short-term changes such as sharp increases or decreases in the number of cases, indicating emerging outbreaks or effects interventions; or long-term shifts, indicating the appearance of the disease in previously unaffected age groups or geographical regions. The literature on statistical analysis of disease surveillance data is vast, and interested readers are recommended to refer to previously published reviews (Brookmeyer & Stroup 2003, Sonesson & Bock 2003, Lawson & Kleinman 2005, Wong & Moore 2006, Burkom 2007) for a more thorough analysis of existing methods.

Most of the algorithms used in disease surveillance are adapted from other fields such as industrial process control or econometrics (Buckeridge et al. 2008), but some have been developed specifically for disease surveillance. Time series methods, mean-regression methods, auto-regressive integrated moving average (ARIMA) models, hidden Markov models (HMMs), Bayesian HMMs, and scan statistics (Pelecanos et al. 2010) are among the most commonly used algorithm classes. The spatial and space-time statistics, specifically, have gained popularity among practitioners as methods for the detection of disease clusters (Kulldorff et al. 2007).

The detection algorithm is chosen based on the needs of the users, and the available data sources. To aid the decision, the performance of aberrancy-detection algorithms are often expressed in terms of sensitivity, specificity, and timeliness (Kleinman & Abrams 2006). Sensitivity (true positive rate) is the probability that an alarm is raised given that an outbreak occurs. Specificity (true negative rate) is the probability that no alarm is raised given that no outbreak occurs. Timeliness is the difference in time between the event and the raised alarm. Additionally, to be able to understand and describe the detection algorithms better, researchers have proposed a classification scheme algorithms that considers the types of information and the amount of information processed by the algorithms: number of accessible data sources, number of covariates in each source, and the availability of spatial information (Buckeridge et al. 2005).

1.4.3 Notification

The results of the analyses are visualised and communicated to the users by the notification component. The simplest method of communication is by providing the statistical output from the analysis method directly to the user, as a table or in plain text. Although the output contains all the essential information, understanding these reports is often time-consuming, and they quickly become overwhelming if many data sources, diagnoses, or geographic regions are involved in the analysis.

The most common way of summarising the results is by displaying their values at different time points, using line charts or bar graphs. The variable may be case reports, ambulance dispatches, drug sales, or any other indicator used in surveillance. If previously computed historical baselines exist, they may also be plotted on the same graph, to put the current results in a larger context. Scatter plots and pie charts may also be used to summarise non-temporal components of the analysis.

When a spatial analysis method is used, the same variable may be displayed using a map where colours, shades, or patterns illustrate the differences between geographical regions (Cromley & Cromley 2009). Results of clustering methods, such as spatial scan statistics, may also be visualised on maps, often using geometric shapes or grids drawn on the map in addition to the regional borders (Boscoe et al. 2003). Visualising the results of hybrid spatio-temporal analysis methods may be achieved by animating the map, or presenting snapshots from the same map at different time points side-by-side.

Alternatively, Geographic Information Systems (GIS) may be used to visualise spatial or spatio-temporal analysis results if the data source contains detailed geographical information. These systems are commonly used in disease surveillance (Nykiforuk & Flaman 2011), and epidemiologists are more likely to be familiar with GIS software given the long tradition of their usage (Clarke et al. 1996). In some cases, GIS include their own analysis tools (Chung et al. 2004), but these may be bypassed by importing the analysis results directly to the visualisation component. A simpler system, Google Earth (Google 2011), also provides similar functionality for spatial visualisation.

The result reports and visualisations are communicated to the users periodically through email, SMS, automated phone calls, web sites, or a dedicated display unit placed at the institution tasked with disease surveillance.

1.5 Implementations of disease surveillance systems

Four disease surveillance systems are presented in this section to illustrate different aspects of existing syndromic surveillance systems. For additional information on other systems, the reader is recommended to refer to Chen et al. (2009), which includes an overview of 50 syndromic surveillance systems and examines eight in further detail. An earlier review of 115 disease surveillance systems, including nine syndromic surveillance systems, by Bravata et al. (2004) is also informative.

1.5.1 BioSense

BioSense is a CDC (Centers for Disease Control and Prevention) initiative that aims to "support enhanced early detection, quantification, and localisation of possible biologic terrorism attacks and other events of public health concern on a national level" (Bradley et al. 2005, p.1). The software component of the initiative is called the BioSense application. The development of the application started in 2003, and the first version was released in 2004.

Initially BioSense included three national data sources: United States Department of Defence military treatment facilities, United States Department of Veterans Affairs treatment facilities, and Laboratory Corporation of America (LabCorp) test orders. In a later technical report, the BioSense data sources were reported to also include state/regional surveillance systems, private hospitals and hospital systems, and outpatient pharmacies (CDC 2008). As of May 2008, 454 hospitals from 26 US states were sending data to BioSense.

The BioSense application classifies incoming data into eleven syndrome categories: botulism-like, fever, gastrointestinal, hemorrhagic illness, localised cutaneous lesion, lymphadenitis, neurologic, rash, respiratory, severe illness and death, and specific infection. The daily statistical analysis is performed using CUSUM (Hutwagner et al. 2003), SMART (Kleinman et al. 2004), and W2 (a modified version of the C2 method (Hutwagner et al. 2003) for anomaly detection. The data reporting component displays the results of the analyses as spreadsheets of observed case counts, time series graphs, patient maps, or detailed case reports.

In 2010 CDC started redesigning the BioSense program. The redesign aims to expand the scope of BioSense beyond early detection to contribute information for "public health situational awareness, routine public health practise, [and] improved health outcomes and public health" (CDC 2011). Earlier presentations about the future of the project have noted additional goals about improving the usability of biosurveillance tools and "reducing excessive features which miss the needs of the users" (Kass-Hout 2009*b*, p.19). Open sourcing of the system is also included as a possibility for the

redesigned BioSense project (Kass-Hout 2009*a*).

The initial motivation for the development and operation of the BioSense application was expressed primarily in terms of preventing biologic terrorism (Bradley et al. 2005). As part of the redesign, the motivation for developing the system is broadened considerably:

> The goal of the redesign effort is to be able to provide nationwide and regional situational awareness for all-hazard health-related threats (beyond bioterrorism) and to support national, state, and local responses to those threats. (CDC 2011)

The BioSense program has also contributed to the International Society for Disease Surveillance report on developing syndromic surveillance standards and guidelines for meaningful use (ISDS 2010). The current BioSense application is one of the largest syndromic surveillance systems in existence, and the scope of its next iteration is likely to be influential in defining what is viable in the field of disease surveillance systems.

1.5.2 RODS

The development of the *Real-Time Outbreak and Disease Surveillance system* (RODS) began in 1999 at the University of Pittsburgh for the purpose of detecting the large-scale release of anthrax (Tsui et al. 2003). The sixth iteration of the software is currently reported to be under development and the source code for several versions licensed under GNU GPL or Affero GPL are available from the RODS Open Source Project website (RODS 2009).

The first implementation of RODS in Pittsburgh, Pennsylvania collected chief complaints data from eight hospitals, classified them into syndrome categories, and analysed the data for anomalies (Espino et al. 2004). The system was then expanded to collect additional data types and deployed in multiple states. It was also used as a user-interface to the American National Retail Data Monitor (Wagner et al. 2003), which collects over-the-counter medication sales. The most recent publicly available version of RODS supports user-defined syndrome categories. Implementations of the recursive least-squared (RLS) algorithm (Hayes 1996) and an initial implementation of the wavelet-detection algorithm (Zhang et al. 2003) are also included. The results of the analyses can be displayed as time series graphs, or work with a GIS to create maps of the spatial distribution.

From a data collection perspective, RODS is the decentralised counterpart of BioSense. Unlike BioSense, which collects data from a large number of sources centrally within a single implementation, RODS is designed to be installed at facilities on the sub-national level to collect and analyse the available data locally. In 2009, more than 300 healthcare facilities in 15 states in the U.S., more than 200 in Taiwan, and an unspecified number in Canada were being monitored by independent RODS implementations (RODS 2009).

At the time of writing, no updates to the RODS open source project have been committed to the code repository for the last two years, and the latest available RODS publications date back to 2008. It is unclear if RODS 6 will be released in the future, but the availability of the source code for many earlier versions makes RODS an important resource for developers of disease surveillance systems.

1.5.3 BioStorm

The BioStorm system (Biological spatio-temporal outbreak reasoning module) has been developed at the Stanford Center for Biomedical Informatics Research in collaboration with McGill University. The goal of the project is to "develop fundamental knowledge about the performance of aberrancy detection algorithms used in public health surveillance" (BioSTORM 2009). The source code for the system is available at BioSTORM (2010).

The aim of the BioStorm project is to create a scalable system that integrates multiple data sources, includes support for many problem solvers, and provides flexible configuration options (O'Connor et al. 2003). The defining feature of the project is the central use of ontologies. A data-source ontology is used to describe data sources. The descriptions are then used to map to suitable analysis methods available in the system's library of problem solvers (Crubézy et al. 2005). Intermediate components such as a data-broker, a mapping interpreter, and a controller are used to connect the data sources to the analysis methods. The use of ontologies is intended to ease the process of adding new data sources and new analysis methods to an existing BioStorm implementation. No existing syndrome categories or visualisation components are provided, but any category or visualiser can be added to the system according to the needs of its users.

The BioStorm project differs from the majority of disease surveillance systems primarily due to its highly complex mechanism for classifying data sources and problem solvers. The developers reflect on the high overhead of this approach, but state that the overhead is acceptable for systems that connect to many data sources and require diverse analysis methods (Buckeridge et al. 2003). In contrast, most of the existing syndromic surveillance systems do not suffer from this overhead, but require additional programming to accommodate new data sources or methods.

The complexity of the BioStorm project creates a significant obstacle for implementation in a public health facility for day-to-day monitoring. However, the feature set provided by the project is ideal for systematically comparing and evaluating the performance of different analysis methods on different data sources. The possibility of categorising not only the syndromes, but also the data sources and the analysis methods (Pincus & Musen 2003) promises to simplify experiment design for evaluating detection algorithms.

The BioStorm source code has not been updated since 2010, and the

most recent publication related to the project dates back to 2009, but the source code continues to be available from the Stanford Center for Biomedical Informatics Research (BioSTORM 2010).

1.5.4 HealthMap

HealthMap is a freely accessible web site that integrates data from electronic sources, and visualises the aggregated information onto the world map, classified by infectious disease agent, geography, and time. The project aims to deliver real-time information for emerging infectious diseases. It has been online since 2006, and its current data sources include Google News, the ProMED mailing list, World Health Organisation announcements and Eurosurveillance publications, among others (HealthMap 2011*a*). HealthMap uses automated text processing to classify incoming alerts and to create or update points of interest on the world map based on the classification results (Freifeld et al. 2008). The time-frame of alerts, the number of alerts, and the number of sources providing information are reflected by the colour of the markers for the points of interest on the world map. HealthMap also includes an interface for users to report missing outbreaks.

HealthMap's reliability, much like any other system, depends on the reliability of its data sources. Since it accesses less reliable sources compared to the systems discussed previously, different weights are assigned to different sources based on their credibility when creating reports to offset the influence of less reliable reports (Brownstein et al. 2008).

HealthMap is unique among the systems discussed so far because its analysis results are available to all world wide web users instead of a small group of experts. The results can be made available without major privacy concerns because all of the incoming data are also publicly available on the web. Another notable system that employs a similar approach is EpiSpider (Tolentino et al. 2007).

A recent HealthMap feature, *Outbreaks Near Me* (HealthMap 2011*b*), provides the users with mobile tools to report and view outbreaks. Accessing the system without a standard browser requires a smart-phone which limits its availability, but it is argued that such limitations will eventually be overcome with cheaper devices (Freifeld et al. 2010). The development of HealthMap-like systems signifies the presence of a different perspective in public health surveillance, where a larger group of users are able to influence the surveillance process and access the results of statistical analyses.

1.6 Discussion

When building tools to improve the health of populations, technical advances in the development of disease surveillance systems ensuring timely detection, less false positives, etc., are clearly important. However, the field of public

health defines its object of interest as a *population*, and issues that directly affect the lives of individuals comprising the population must be considered in a broader perspective when developing surveillance systems.

Disease surveillance is used to monitor the population for signs of disease, and, in case of detection, to propose strategies to cure or control it. It functions at a different level, away from the population itself, watching for signs of the disease hosted in the population. Fearnley (2010) provides a detailed account of this conceptual decoupling of the disease and the host, in the contemporary understanding of disease surveillance, through the career of the influential epidemiologist Alexander Langmuir. In his examination, Fearnley locates the transformation of the epidemic "from a problem of population pathology into a discrete event framed by outbreak and subsidence" (p.42). The decoupling, now widely accepted as a valid methodology for disease surveillance, is carried one step further with the introduction of syndromic surveillance. In syndromic surveillance, data streams tracking the activities of the population are monitored for unusual signs associated with categories that correspond to diagnoses, which may in turn indicate the presence of actual diseases. Predictably, two levels removed from the source, any indicator becomes weaker. Practitioners have voiced the concern that syndromic surveillance signals are insignificant in the absence of follow-up investigations (Heffernan et al. 2004*b*, p.863).

The ideological shift, following the methodological one, that syndromic surveillance has offered to the practise of disease surveillance is identified by Fearnley (2008) in the conflict between two *styles* of governing: "public health (a responsibility for maximal population health) and preparedness (a concern for disaster-scale events)" (p.1615). Public health as a governing style aims to increase the health of the governed population while acknowledging that the scope of its acts are limited by both costs and available knowledge, or the perceived lack of it. This style "uses legal authority to expand its access to population health data" (p.1617). In contrast, the roots of the style of preparedness lie in the Cold War era, where the distinctions between battlefield and homefront were blurred, and an awareness of the permanent state of readiness where threats can attack anywhere without warning was encouraged. The techniques of the preparedness style involve declaring structures or institutions vulnerable by imagining the effects of a threat materialising or being carried out successfully in the future. Due to its positioning against uncertainty, or towards the prevention of the uncertain "[t]he evaluation of syndromic surveillance for bioterrorism preparedness could not make reference to a statistical logic of costs and benefits"(p.1627). It is impossible to judge the costs or who would suffer them, or conversely, the benefits and who would enjoy them, without specifying the properties of the unknown to be prepared for.

In an attempt to frame the unknown, the motivation for developing syndromic surveillance systems often includes the rapid detection and pre-

vention of acts of bio-terrorism. However, no bio-terrorism attacks anywhere in the world were detected in the first half of the last decade (Cooper et al. 2006), and none have been reported in the second half. Researchers had identified the threat of bio-terrorism as an exaggeration (Sidel et al. 2002) soon after the development of national syndromic surveillance systems in the United States. Two years later, they asked the public health community to "acknowledge the substantial harm that bioterrorism preparedness has already caused and develop mechanisms to increase our public health resources and to allocate them to address the world's real health needs." (Cohen et al. 2004, p.1670). A later assessment of the bio-terrorism threat also reached similar conclusions (Leitenberg 2005).

When the diversity of communicable diseases hosted by the world population and the suffering caused by the diseases are considered against the threat of bio-terrorism, the rift between the two governing styles, public health and preparedness, is clearly visible. Proposals for developing new disease surveillance systems must either engage the preparedness question and clearly identify the goals of surveillance, or risk searching endlessly for the significant in a sea of noise.

1.7 Advances on state-of-the-art

The structural analysis of syndromic surveillance systems presented earlier provides tools to plan for the development of new systems, or to aid the understanding of existing systems. The first paper in chapter 2 describes a design process that has been guided by these principles. The design was inspired by syndromic surveillance, but its solutions are aimed towards a later stage in disease surveillance, after the diagnoses are reported. The freely available, open source software package aims to ease the burden of connecting a data source of reported cases to multiple statistical analysis methods, and to provide a communication channel for regular updates of the results to epidemiologists.

The second paper presents the lessons learned from a software development project of more than 100 person months in the form of a check list. The open source software package, a spatially explicit model for the entire Swedish population built on registry data, has been used to simulate outbreaks of smallpox and influenza-like illnesses. Computational models are used in disease surveillance systems to create simulated data for testing detection algorithms, but using the simulation results for decision support, the main goal of the project, introduces new methodological challenges. The discussion of these challenges contributes to the methodological advancement of computational epidemiology.

Complex disease surveillance systems are still in their infancy. This thesis explores their foundations, analyses the structure of existing examples,

and offers guidelines for future research in the field.

1.8 Author's contributions

The next chapter contains two papers with Cakici as the main author. In total, Cakici has contributed approximately 27 person months to the development of the described software packages.

The first paper, Cakici et al. (2010), was initiated by Cakici, Saretok, and Hulth as the project leader. Saretok had already built a prototype before Cakici joined the project. Cakici and Saretok re-designed and re-developed the application using a database for storage instead of local files to ensure scalability. The first draft of the manuscript was prepared by Cakici, Hulth and Saretok. Cakici and Hulth were responsible for editing the manuscript, and it was submitted by Cakici.

The second paper, *A workflow for software development within computational epidemiology* (under review, Journal of Computational Science), was produced in close collaboration with Boman. The writing and editing of the text was shared equally, and the simulations were set up and analysed by Cakici.

The research resulting in this licentiate thesis began in 2009 at the Swedish Institute for Communicable Disease Control (SMI), and SICS, the Swedish Institute of Computer Science. From January 2011, it continued at the Royal Institute of Technology (KTH), the unit for Software and Computer Systems (SCS) at the School of ICT.

1.9 A list of disease surveillance systems

A list of disease surveillance systems and publications describing them are included below for interested readers.

System acronym	Name or description	Reference
B-SAFER	Bio-surveillance analysis, feedback, evaluation and response	(Brillman et al. 2003)
BioPortal	An information sharing and data analysis environment	(Zeng et al. 2005)
BioSense	A national early event detection and situational awareness system	(Bradley et al. 2005)
BioSTORM	Biological spatio-temporal outbreak reasoning module	(Crubézy et al. 2005)
btsurveillance	The national bioterrorism syndromic surveillance demonstration program	(Yih et al. 2004)
DiSTRIBuTE	Influenza surveillance system	(Diamond et al. 2009)
EARS	Early aberration reporting system	(Hutwagner et al. 2003)
ESSENCE II	The electronic surveillance system for the early notification of community-based epidemics	(Lombardo et al. 2003)
HealthMap	Global health, local information	(Brownstein et al. 2008)
INFERNO	Integrated forecasts and early enteric outbreak detection system	(Naumova et al. 2005)
RODS	Real-time outbreak detection system	(Tsui et al. 2003)
RSVP	Rapid syndrome validation project	(Zelicoff et al. 2001)
AEGIS	Automated epidemiologic geotemporal integrated surveillance system	(Reis et al. 2007)
CASE	Computer assisted search for epidemics	(Cakici et al. 2010)
EWRS	Early warning and response system	(Guglielmetti et al. 2006)
NEDSS	The national electronic disease surveillance system	(M'ikantha et al. 2003)
NNDSS	Australian notifiable disease surveillance system	(*NNDSS* 2010)
SmiNet	An internet-based surveillance system for communicable diseases in Sweden	(Rolfhamre et al. 2006)
TESSy	The European surveillance system	(ECDC 2010)

Chen et al. (2009) describe the first 12 systems listed above in further detail. As noted previously, Bravata et al. (2004) provide an extensive review of 115 disease surveillance systems.

Chapter 2

Papers

2.1 CASE: a framework for computer supported outbreak detection

Cakici *et al. BMC Medical Informatics and Decision Making* 2010, **10**:14
http://www.biomedcentral.com/1472-6947/10/14

BMC
Medical Informatics & Decision Making

SOFTWARE Open Access

CASE: a framework for computer supported outbreak detection

Baki Cakici*[1,2], Kenneth Hebing[1], Maria Grünewald[1], Paul Saretok[1] and Anette Hulth[1]

Abstract

Background: In *computer supported outbreak detection*, a statistical method is applied to a collection of cases to detect any excess cases for a particular disease. Whether a detected aberration is a true outbreak is decided by a human expert. We present a technical framework designed and implemented at the Swedish Institute for Infectious Disease Control for computer supported outbreak detection, where a database of case reports for a large number of infectious diseases can be processed using one or more statistical methods selected by the user.

Results: Based on case information, such as diagnosis and date, different statistical algorithms for detecting outbreaks can be applied, both on the disease level and the subtype level. The parameter settings for the algorithms can be configured independently for different diagnoses using the provided graphical interface. Input generators and output parsers are also provided for all supported algorithms. If an outbreak signal is detected, an email notification is sent to the persons listed as receivers for that particular disease.

Conclusions: The framework is available as open source software, licensed under GNU General Public License Version 3. By making the code open source, we wish to encourage others to contribute to the future development of computer supported outbreak detection systems, and in particular to the development of the CASE framework.

Background

In this paper, we describe the design and implementation of a *computer supported outbreak detection system* called CASE (named after the protagonist of the William Gibson novel Neuromancer), or Computer Assisted Search for Epidemics. The system is currently in use at the Swedish Institute for Infectious Disease Control (SMI) and performs daily surveillance using data obtained from *SmiNet* [1], the national notifiable disease database in Sweden.

Computer supported outbreak detection is performed in two steps:

1 A statistical method is automatically applied to a collection of case reports in order to detect an unusual or unexpected number of cases for a particular disease.

2 An investigation by a human expert (an epidemiologist) is performed to determine whether the detected irregularity denotes an actual outbreak.

The main function of a computer supported outbreak detection system is to warn for *potential* outbreaks. In some cases, the system might be able to detect outbreaks earlier than human experts. Additionally, it might detect certain outbreaks that human experts would have overlooked. However, the system does not aim to replace human experts (hence the prefix "computer supported"); it should rather be considered a complement to daily surveillance activities. To a smaller extent, the system can also aid less experienced epidemiologists in identifying outbreaks.

Systems for outbreak detection which support multiple algorithms include RODS [2], BioSTORM [3] and AEGIS [4]. Additionally, computer supported outbreak detection systems operating on the national level have been used previously in a number of countries, including Germany [5] and the Netherlands [6].

Health care in Sweden

The health care system in Sweden is governed by 21 county councils. Each county has appointed a medical officer, who is in charge of the regional infectious disease prevention and control. Every confirmed or suspected

* Correspondence: baki.cakici@smi.se

[1] Swedish Institute for Infectious Disease Control (SMI), 171 82 Solna, Sweden

case of a notifiable disease is reported both to the county medical officer and to SMI. At SMI, the regular national surveillance is currently performed by thirteen epidemiologists, each in charge of a number of different diseases.

All 21 county medical officers as well as the majority of the hospitals and the laboratories in Sweden are connected to the SmiNet database. The database collects clinical reports and information on laboratory verified samples. In 2008, a total of 174 811 reports were submitted to SmiNet. 87 per cent of these reports were submitted electronically and those that were not submitted electronically were entered into SmiNet manually. Of the 92 744 lab reports, as much as 97 per cent were submitted electronically and 62 per cent fully automatically. The reports were subsequently merged into 74 367 case reports. These reports form the basis of the data used by CASE to perform outbreak detection.

Implementation

CASE is designed to be administered using a graphical interface, and can operate on all of the 63 notifiable diseases in Sweden. One or more statistical detection methods can be applied to each disease. If more than one method is activated, result reports are generated independently. By default, the data are aggregated over all disease subtypes, but the system allows detection of single subtypes as well. When an outbreak signal is generated, an alert is sent by email to all members of the notification list for that particular disease.

CASE is composed of three interconnected components for *configuration*, *extraction* and *detection*. The configuration component provides a graphical user interface for modifying detection parameters and editing the list of recipients for generated alerts. The extraction component is used to copy data from the national case database to the local database. The detection component is scheduled to run at regular intervals and automatically applies the chosen statistical methods to the currently selected diseases.

System Description

CASE is developed using Java to ensure platform-independence of all components. Currently at SMI all three components run on Ubuntu, a Linux-based operating system. The local database for CASE is MySQL and the national database, SmiNet, is Microsoft SQL Server 2005.

Figure 1 shows the flow of information within the framework. The extraction and detection components are scheduled to run once every 24 hours at midnight using the standard Unix scheduling service *cron*. When the extraction component is executed, it transfers data from SmiNet to the local database. The local database stores the case data and the configuration parameters for all algorithms. The configuration module can be used to view and modify the parameters. The detection component is executed automatically after all required data have been extracted from SmiNet. It applies the detection methods with the given parameters to the case data for the selected diseases, and emails notifications if any alerts are generated. Detailed logs of these processes are generated automatically.

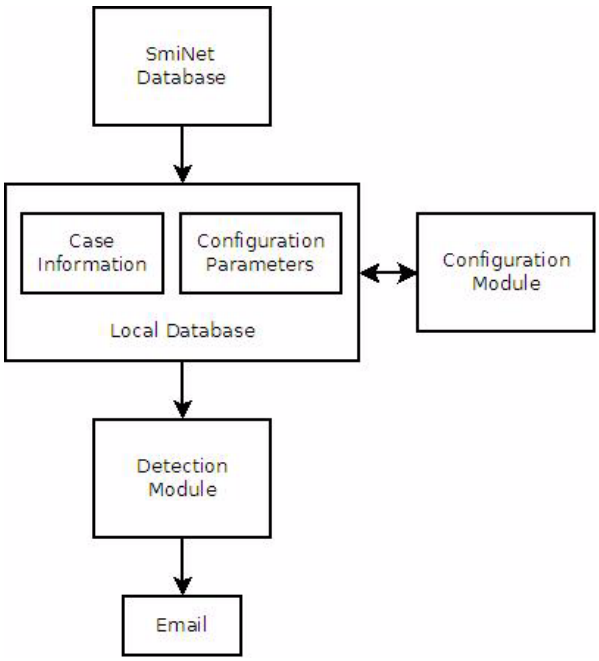


Figure 1 CASE Flowchart. A flowchart demonstrating the detection process in CASE.

Configuration

The configuration component is a graphical user interface that allows the administrator to mark diseases for detection, choose the detection methods to be applied to each diagnosis/subtype and manage the list of epidemiologists that will receive alerts in case a warning is generated. The settings are stored in a local database that is also accessed by the other two components. The system can be administered by multiple users who access the same local database.

Figure 2 shows a screenshot of the graphical user interface for the CASE administrator. The notifiable diseases are displayed in the left column. These entries can be expanded using the arrow to display their subtypes. Parameters for the current selection are shown on the right hand side. The *Algorithms* tab lists the available methods. Parameters for the selected method can be modified by double-clicking the name of the method. The *E-mail* tab contains a list of recipients for the selected disease and/or subtype. If an alert is generated after detection, the algorithm that generated the alert is high-

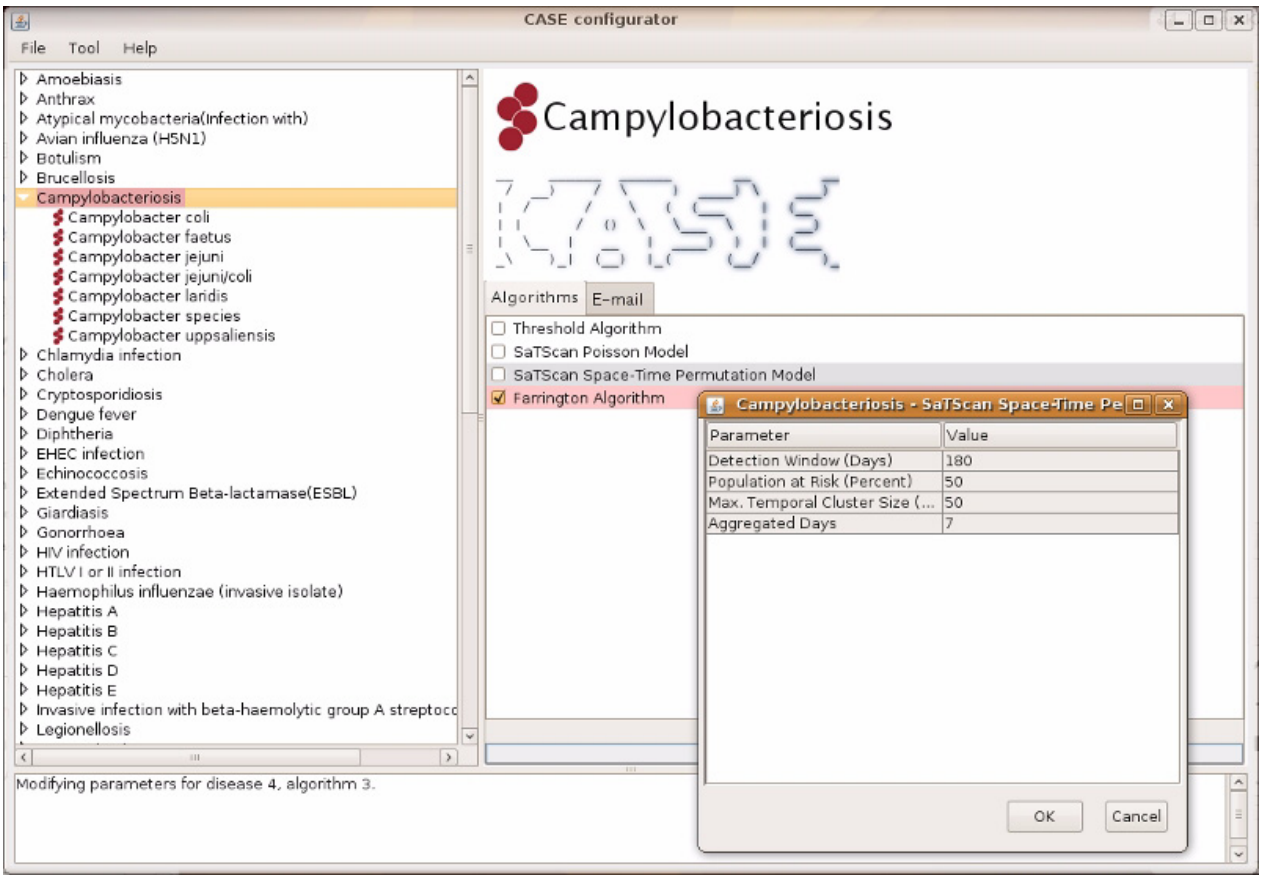


Figure 2 Administrator GUI. A screenshot of the graphical user interface for the CASE administrator.

lighted in red. The flag is automatically cleared every night before a new detection batch is executed.

Extraction

CASE uses data retrieved from SmiNet to perform outbreak detection. A case report is created in SmiNet when a clinical or a laboratory report is received, provided that this patient does not already exist in the database. When additional reports arrive, the original case report is automatically updated with the new information. Depending on the number of days that have elapsed since the last time a patient received a particular diagnosis, a new case report might be created for the same diagnosis and patient. For a detailed technical description of SmiNet, see [1].

The extraction component populates the local database with data from the case reports stored in SmiNet. Diagnosis, lab species, date, and reporting county are copied for every case, except those with infections that are reported to have originated abroad. No information that can reveal a patient's identity is used in the outbreak detection process. There are approximately twenty dates in SmiNet for each case report, ranging from dates that are automatically generated by the system to dates entered by the clinician or the laboratory. There is, however, only one date that is available on all case reports, namely *statistics date*. This automatically set date corresponds to when a patient first appears in SmiNet with a particular diagnosis. The date that would best reflect when a patient fell ill is the date when the sample was taken from the patient. However, many case reports do not contain this date. For example, for 2008 this date is missing in 29 per cent of the case reports. When the case information is copied from SmiNet to the local database, the extraction component fetches the statistics date as the date for the case.

Detection

CASE is developed by the Swedish Institute for Infectious Disease Control, and has a national perspective on outbreaks. Its primary role is to find outbreaks that cover more than one county, especially those with few cases in each affected county, as these might be difficult to detect for the local authorities.

The detection component uses the selected statistical method(s) on all activated diseases and sends notification emails if any alerts are raised. If there are too few data points for a detection algorithm to produce a result --

which is often the case for detection on the subtype level -- this information is written to the log file. The system currently supports four different statistical methods for detection: SaTScan Poisson [7], SaTScan Space-Time Permutation [8], an algorithm developed by Farrington et al. [9], and a simple threshold algorithm. The methods are briefly described below. Three of the four methods are freely available implementations, while the fourth was developed within the project and is included in CASE's source code. For the external programs, input generators and output parsers are also contained within the source code. It is possible to extend the system with additional statistical methods, although this requires a certain familiarity with the Java programming language. We are currently in the process of adding the *OutbreakP* method [10] to the core package.

SaTScan is a freely available spatial, temporal and space-time data analysis platform [11]. Two algorithms from this application are used in CASE: *SaTScan Poisson* which uses the discrete Poisson SaTScan model to search for spatial clusters and *SaTScan Space-Time Permutation*, which searches for spatio-temporal clusters. Both models are applied to data at the county-level resolution. The population data required by SaTScan Poisson are obtained from Statistics Sweden [12]. The SaTScan Poisson parser, developed specifically for CASE, raises an alert if a detected cluster ends within the last week.

The third detection method was developed by, and is in regular use at the Health Protection Agency in England and Wales [9]. In CASE, we use the surveillance R-package implementation [13] of the method and we refer to it as the *Farrington algorithm*. The algorithm is used on data aggregated at the national level, to investigate if the current disease incidence exceeds that of the reference data from previous years. The CASE parser for the Farrington output ensures that an alert is sent only if an exceedance occurred during the last two weeks. The required window size is implemented as a sliding window of seven days and detection is performed daily.

The *threshold algorithm* is used to generate alerts when the number of cases for a particular disease rises above a manually defined value, with the number of cases aggregated at the national level.

For all methods, as long as an outbreak is ongoing according to the results of the statistical analysis, a new alert is raised every night. Figure 3 shows an alert email that is sent to the recipients of "MRSA infection". The graph is automatically generated by the detection component and shows all computed alarms on the x-axis. The computed threshold is denoted by the blue curve (the graph in Figure 3 was generated using simulated data). The email also includes a brief description of the algorithm that generated the alarm.

Results and Discussion

CASE is a technical framework designed to ease the process of connecting a data source with reported cases to various statistical methods requiring different input formats. When using CASE, the user can select the methods that are best suited to the characteristics of a particular disease.

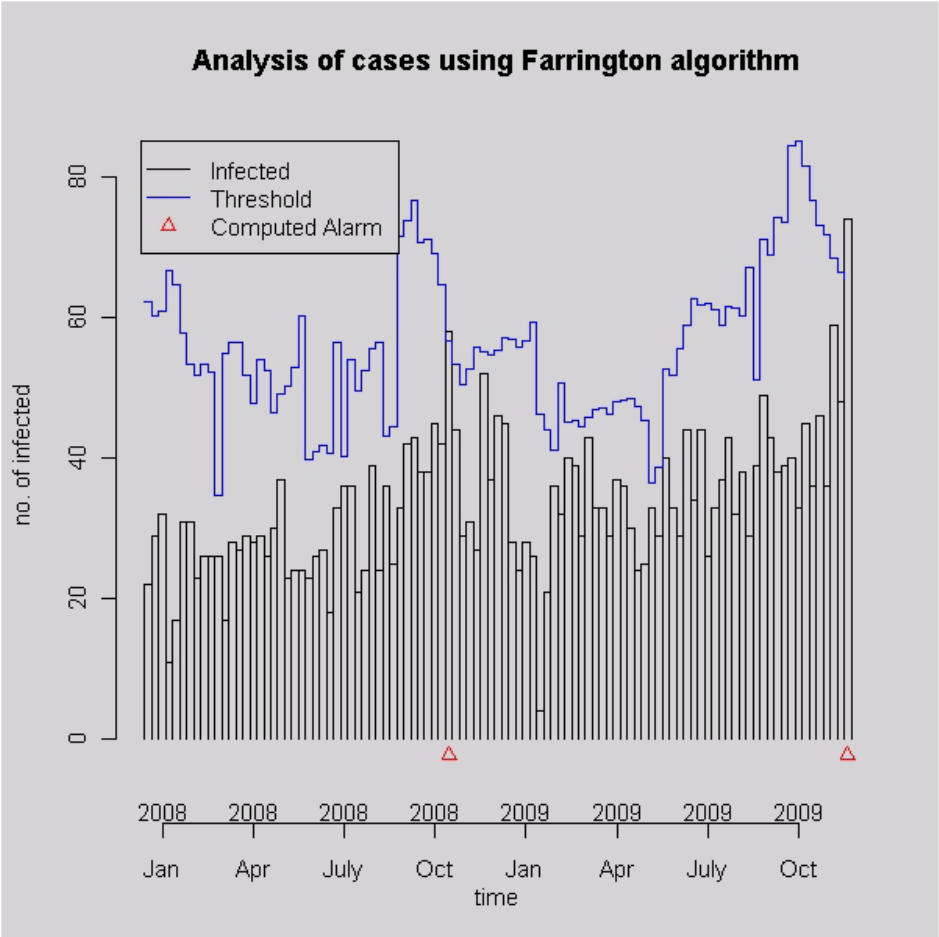
CASE can also be used as a platform for comparing different detection algorithms, although that is not its primary purpose. Since all algorithms use the same data, running multiple detection methods on the same disease regularly and comparing the successful detections and the false warnings can provide insights into the accuracy of a certain method for a given disease. Comparisons and evaluations of the statistical methods currently included in CASE can be found in, for example, [14] and [15]. Here, the importance of calibrating the parameters for the detection methods must be emphasized, something which is still an ongoing work at SMI.

At present, the evaluation of the system is mainly qualitative, consisting of frequent discussions between the epidemiologists and the CASE developers. There is, however, a need for more systematic evaluations of the system, including a questionnaire assessing the users' experience, in addition to quantitative evaluations of the performance of the algorithms and the parameter settings. To facilitate the quantitative evaluations, we plan to extend the functionality of CASE to incorporate an evaluation module allowing the algorithms to be run retrospectively, with analysis carried out for each day in a specified time period. The main objective is not a general comparison of the algorithms, but an assessment of their performance in the specific context of the data they are used on. Where external data telling when actual outbreaks have occurred are available, measures such as sensitivity and specificity can be calculated. The evaluation module would provide valuable guidance in the choice of algorithms and parameter settings for the end user. Another evaluation feature we consider implementing is the possibility to run simulated data in the system.

CASE currently uses emails for notification. The advantage of this approach is that it presents information to the users in a familiar way and does not require them to learn how to operate a new interface. The disadvantage, on the other hand, is that the system becomes one-sided if the emails do not include a feedback mechanism. Regardless of the actual implementation, a system for providing feedback from the receivers of the signals is essential. Currently, users who would like to provide feedback on CASE output are instructed to email the administrator.

As expected, a relatively simple method operating on accurate and informative data produces better results than a complex method operating on noisy data. There-

The algorithm "Farrington" has been applied to **MRSA infection** and has detected a signal that could signify an outbreak.



The graph in this email shows at what point(s) in time the threshold has been exceeded.

How does the Farrington algorithm (Farrington et al, 1996) work?
The algorithm compares the current number of cases to a threshold computed from data from previous years. A signal is given when data differ from what is expected in a statistically significant way, for at least one of the last two weeks. The significance level can be altered by the CASE administrator. There may be both outbreaks not detected (false negatives) and a signal when there is no outbreak (false positives).

Farrington CP, Andrews NJ, Beale AD, Catchpole MA, A statistical algorithm for the early detection of outbreaks of infectious disease. Journal of the Royal Statistical Society. Series A, 159:547-563, 1996.

For further information send an email to: case@smi.se

Figure 3 Alert Email. A sample email for a disease alert.

fore, the most important factor for creating a reliable outbreak detection system is to ensure the quality of the input data. If the input is not reliable, improving the data collection process from local medical centres is a much better investment than trying to perform automatic detection on inaccurate data. Additionally, expectations from an automated detection system must be realistic. For a computer, detecting ongoing outbreaks and sea-

sonal regular outbreaks is possible, but predicting an outbreak at onset is currently not feasible.

CASE is designed primarily to analyze case reports and does not provide syndromic surveillance support using external data sources, unlike RODS [2] or BioSTORM [3]. The only requirement for the operation of CASE is access to a case database for notifiable diseases. All scripts to create and configure the intermediate local database are included in the software package. The local database is used to selectively copy and store case reports after removing all information that can reveal a patient's identity. We believe that the ease of configuration and maintenance in addition to the possibility of operating without storing highly sensitive data make CASE a strong candidate for use in national infectious disease surveillance.

Conclusions

In this paper we have described the design and implementation of a publicly available technical framework for computer supported outbreak detection. The source code is licensed under GNU GPLv3 [16] and is available from https://smisvn.smi.se/case.

The CASE framework is designed to be a complete system for computer supported outbreak detection at the national level. We are aware that any outbreak detection system must always be adapted to a particular context, where national requirements and regulations will affect the implementation of the system. Such adaptations can easily be made within the described framework. By making the code open source, we wish to encourage others to contribute to the future development of computer supported outbreak detection systems, and in particular to the development of the CASE framework.

Availability and requirements

The source code for CASE is licensed under GNU General Public License Version 3 (GPLv3), and is available for download from https://smisvn.smi.se/case. The provided documentation and the interface are written in English. The following software must be installed on the target system in order to use CASE:

- Linux or Windows operating system that can run Sun Java Runtime Environment 6.0 (or higher)
- MySQL 5.1 (or higher)
- SaTScan version 8.0.1 (or higher)
- R version 2.9.1 (or higher)
- ImageMagick 6.5.4 (or higher)

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AH, BC and PS designed and developed the CASE framework. BC, KH and PS implemented the framework. KH and MG worked on improving the application. AH and BC drafted the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We would like to thank the epidemiologists at SMI, especially Margareta Löfdahl and Tomas Söderblom, both enthusiastic recipients of the notifications during the early stages of the project. We also thank Martin Camitz for naming the system. Finally, we would like to thank everyone who provides reports for and works with SmiNet. The project is funded by the Swedish Civil Contingencies Agency (formerly the Swedish Emergency Management Agency).

Author Details

[1]Swedish Institute for Infectious Disease Control (SMI), 171 82 Solna, Sweden and [2]Royal Institute of Technology (KTH), 100 44, Stockholm, Sweden

Received: 11 September 2009 Accepted: 12 March 2010
Published: 12 March 2010

References

1. Rolfhamre P, Janson A, Arneborn M, Ekdahl K: **SmiNet-2: Description of an internet-based surveillance system for communicable diseases in Sweden.** *Euro Surveill* 2006, **11(5):**626.
2. Tsui FC, Espino JU, Dato VM, Gesteland PH, Hutman J, Wagner MM: **Technical description of RODS: a real-time public health surveillance system.** *J Am Med Inform Assoc* 2003, **10(5):**399-408.
3. Crubezy M, O'Connor M, Buckeridge DL, Pincus Z, Musen MA: **Ontology-Centered Syndromic Surveillance for Bioterrorism.** *IEEE Intell Syst* 2005, **20(5):**26-35.
4. Reis BY, Kirby C, Hadden LE, Olson K, McMurry AJ, Daniel JB, Mandl KD: **AEGIS: A Robust and Scalable Real-time Public Health Surveillance System.** *J Am Med Inform Assoc* 2007, **14(5):**581-588.
5. Krause G, Altmann D, Faensen D, Porten K, Benzler J, Pfoch T, Ammon A, Kramer MH, Claus H: **SurvNet electronic surveillance system for infectious disease outbreaks, Germany.** *Emerg Infect Dis* 2007, **12(10):**1548-55.
6. Widdowson MA, Bosman A, van Straten E, Tinga M, Chaves S, van Eerden L, van Pelt W: **Automated, Laboratory-based System Using the Internet for Disease Outbreak Detection, the Netherlands.** *Emerg Infect Dis* 2003, **9(9):**1046-52.
7. Kulldorff M: **A spatial scan statistic.** *Commun Stat Theory Methods* 1997, **26:**1481-1496.
8. Kulldorff M, Hartman Heffernan J, Assunção R, Mostashari F: **A Space-Time Permutation Scan Statistic for Disease Outbreak Detection.** *PLoS Med* 2005, **2(3):**e59.
9. Farrington CP, Andrews NJ, Beale AD, Catchpole MA: **A statistical algorithm for the early detection of outbreaks of infectious disease.** *J Roy Stat Soc Stat Soc* 1996, **159(3):**547-563.
10. Frisén M, Andersson E, Schiöler L: **Robust outbreak surveillance of epidemics in Sweden.** *Stat Med* 2009, **3:**476-493.
11. **SaTScan - Software for the spatial, temporal, and space-time scan statistics** [http://www.satscan.org]
12. **Statistics Sweden** [http://www.scb.se]
13. Höhle M: **surveillance: An R package for the monitoring of infectious diseases.** *Comput Stat* 2007, **22(4):**571-582.
14. Rolfhamre P, Ekdahl K: **An evaluation and comparison of three commonly used statistical models for automatic detection of outbreaks in epidemiological data of communicable disease.** *Epidemiol Infect* 2005, **134(4):**863-871.
15. Aamodt G, Samuelsen SO, Skrondal A: **A simulation study of three methods for detecting disease clusters.** *Int J Health Geogr* 2006, **5(15):**.
16. **Free Software Foundation - Licenses** [http://www.gnu.org/licenses/gpl.html]

Pre-publication history

The pre-publication history for this paper can be accessed here:
http://www.biomedcentral.com/1472-6947/10/14/prepub

doi: 10.1186/1472-6947-10-14
Cite this article as: Cakici *et al.*, CASE: a framework for computer supported outbreak detection *BMC Medical Informatics and Decision Making* 2010, **10**:14

2.2 A workflow for software development within computational epidemiology

A workflow for software development within computational epidemiology

Baki Cakici[a,b,*], Magnus Boman[a,c]

[a] *Royal Institute of Technology (KTH/ICT/SCS), SE-16440 Kista, Sweden*
[b] *Swedish Institute for Communicable Disease Control (SMI), SE-17182 Solna, Sweden*
[c] *Swedish Institute of Computer Science (SICS), SE-16429 Kista, Sweden*

Abstract

A critical investigation into computational models developed for studying the spread of communicable disease is presented. The case in point is a spatially explicit micro-meso-macro model for the entire Swedish population built on registry data, thus far used for smallpox and for influenza-like illnesses. The lessons learned from a software development project of more than 100 person months are collected into a check list. The list is intended for use by computational epidemiologists and policy makers, and the workflow incorporating these two roles is described in detail.

Keywords: Policy making, computational epidemiology, workflow, individual-based simulation.

1. Introduction

1.1. Computational Epidemiology

In 1916, Ross noted that mathematical studies of epidemics were few in number in spite of the fact that "vast masses of statistics have long been awaiting proper examination" (page 205, [1]). In the 90 years which followed, the studies made were analytic, and the micro-level data available were largely left waiting, to leave room for systems of differential equations built on homogeneous mixing. This is remarkable not least because the modeling problem remains the same throughout history: "One (or more) infected person is introduced into a community of individuals, more or less susceptible to the disease in question. The disease spreads from the affected to the unaffected by contact infection. Each infected person runs through the course of his sickness, and finally is removed from the number of those who are sick, by recovery or by death. The chances of recovery or death vary from day to day during the course of his illness. The chances that the affected may convey infection to the unaffected are likewise dependent upon the stage of the sickness." (page 700, [2]). Heterogeneity is present already in this classic description, in several places; susceptibility, morbidity, and also contact patterns, if only implicitly. Only with the advent of powerful personal computers, were micro-level

*Corresponding author
Email addresses: cakici@kth.se (Baki Cakici), mab@kth.se (Magnus Boman)

data given a role in the modeling of epidemics. Executable simulation models in which each individual could be modeled as an active object with its own attributes [3], often referred to as an agent, began to appear [4, 5, 6]. A new area within computer science, computational epidemiology, has recently become established as the scientific study of all things epidemiological except the medical aspects. This area is turning into computational science (see, e.g., [7]), following the example of computational biology, computational neurology, computational medicine, and several other new areas focusing on building computationally efficient executable models. This development also includes the social sciences, as in computational sociology [8].

1.2. Model Description

The model on which the analysis below is based has been continuously developed since 2002 by a cross-disciplinary group of researchers from the fields of medicine, statistics, mathematics, sociology and computer science. Since 2004, a team of developers have implemented various versions of a software tool, representing the computational part of the model, recently made available as open source software and licensed under GNU General Public License Version 3 [9]. In parallel with the implementation, the requirements on the model have changed many times. It began as a model for predicting the effects of a possible smallpox outbreak in Sweden [10], which was later transformed into a model for studying pandemic influenza, and is now a model that could be used for many different kinds of communicable disease studies (excluding vector-borne diseases, i.e., diseases with animal reservoirs). The model is a detailed representation of real situations, sometimes referred to as a tactical model, as opposed to simpler strategic models [11]. For instance, the model was recently used to study a fictitious scenario of H4N6: a new influenza virus strain that was assumed to be deadly, highly contagious, and introduced into a completely susceptible population. In all, the development project has included more than 100 person months of implementation work, and consists of more than 5000 lines of C^{++} code.

The parameters used to represent individuals in the model are age, sex and current status (alive or deceased). Each individual is also assigned a home, a workplace, and a department within that workplace. The movement of individuals outside of home and workplace are represented using travel status (home or in another location), emergency room visits, and hospitalizations.

Infections caused by social contact outside of work or home are classified as context infections. When the context infection process is active, there is a probability that an infectious individual will infect those that live within a fixed radius. Context contact radius defines the size of neighborhoods, mirroring the interaction of every individual with others, based on geographical proximity and the social network.

The disease affects every individual through three parameters: infectiousness, death risk, and place preference. The infectiousness parameter influences the probability that the infected individual will infect others in the same home, workplace, or neighborhood. The death risk depends on the disease level and is expressed as a probability. Place preference is the probability distribution used when deciding where the individuals will spend their day (workplace, home, primary care, or hospital). These parameters are defined for five levels of sever-

ity: asymptomatic, mild, intermediate, severe, and critical. In addition, there are four disease profiles: asymptomatic, mild, typical, and atypical.

The model description is combined with Swedish data on workplaces, households, and individuals. Workplaces include companies, schools, healthcare, and other state institutions. For each workplace, the data indicate the total number of workers, geographical coordinates, and workplace type. The current version of the simulation platform uses data from the Swedish Total Population Register, the Swedish Employment register, and the Geographic Database of Sweden (cf. [12]).

Because the model was developed with the purpose of being run with data for the country of Sweden, it has been used solely for studying outbreaks in that country. Sweden has relatively many infection clinics and good international reputation for detailed clinical reports of communicable disease. Thus, in some areas of disease control, Sweden works well as a role model. Other countries face special local problems, however, and results have sought to be generalizable, for example contributing to the complicated model of EU care-seeking behavior. Generally speaking, the project goals have included to sensitize policy makers to the scope of possible disruption due to a newly emergent disease event, and to identify a range of policy handles which can be used to respond to such an episode.

A sample case description illustrates how an experiment would be described using the executable model. The sample case simulates the effects of pandemic influenza in Sweden, without any interventions, for 300 days. The simulation is initiated with 50 infected individuals, randomly selected from the entire population. Since the data set is registry data for the entire country, any random selection procedure is uniform, i.e., an individual has a 50 in nine million chance of being initially infected. This does not mirror realistic spread, which would more typically be an airplane or a boat arriving to Sweden with one or more infected individuals on board, but in the sample case it at least provides an opportunity to discuss the complex matter of how epidemics start. The maximum size for an office is set to 16 individuals and all workplaces with more than 16 employees are split into departments, each containing 16 or fewer members. This value is not arbitrary, but corresponds to the average size of a Swedish workplace. Context contacts—the parameter representing the average number of contacts outside the home or the workplace—is set to 15. Even if that number was recommended by the sociologists in the project, it is somewhat arbitrary, and is therefore subjected to sensitivity analyses in our sample case. Naturally, such analyses would be extensive in a real policy case; here the reason for their inclusion is chiefly pedagogical.

1.3. Disposition

A report on lessons learned from the software development project constitutes the bulk of the analysis below. It starts with a description of the workflow in a computational epidemiology project, and observations on the micro-meso-macro link follow. More detailed descriptions of what it actually means to manage and run a simulator are then provided, before discussing the scientific merits and challenges of this kind of research, and the concluding check list is presented.

2. Workflow

2.1. Model Development

The process of developing a model for outbreaks today often includes the development of a simulator, allowing for scenario execution and relatively swift sensitivity analyses. The simulator does not capture the entire model, but only those parts that are subject to uncertainty or those that involve stochastic parameters. The instigator is typically a policy maker (PM), knowledgeable in public health issues, and seeking to evaluate various scenarios. The PM may well have medical training, or even be an epidemiologist. The implementer of the simulator is a computational epidemiologist (CE): a modeler knowledgeable in computer science, and the social sciences, typically without much medical training. Naturally, both PM and CE could denote a team instead of a single person. A schematic workflow for developing and using a simulator, depicting the roles of both PM and CE, is presented in Figure 1.

As in all development projects, work begins with a requirements specification, to which the PM contributes user requirements and the CE contributes technical expertise. From this specification, the simulator is built. It consists of a software package with two parts: a simulation engine and a world description. The latter is not the complete description of the world under study, but covers only those parts that have a bearing on the executable model. This modeling work is carried out by the CE, with considerable assistance from medical professionals. The CE implements the simulator in accordance with the specification and medical expertise. The CE will also seek to verify the accuracy of the simulator (e.g., through extensive testing, or even logical proof). The CE works in two distinct sequential steps that cannot be combined: design and implementation. Software engineers are taught not to modify their design during the implementation stage to "improve" the model, no matter how tempting this might be. If design decisions leak into the implementation stage, the software project quickly becomes impossible to maintain. What software design means in the area of computational epidemiology is the craft of knowing which parameters to vary, being aware of their mutual dependence, and how to openly declare all simplifying assumptions.

Once the simulator is complete it is given a version number, and one may proceed to experiments. For an experiment to be meaningful, the PM must envisage scenarios. The PM must also provide values for some input parameters. Each parameter in the model is important, and even slight changes to an input value might have a drastic effect on the output. The kind of model considered here is a complex system: a system which cannot be understood through understanding its parts. Before the CE can run the system, the world description must be populated with data, which typically need a significant amount of post-processing to allow for smooth use in the simulator. In addition, one must then attempt to ascertain that the resulting data set is accurate and noise-free. The data set in the here described model was sensitive with respected to personal integrity, as it consisted of registry data on the entire Swedish population of approximately nine million individuals. This sensitivity rendered many kinds of replication experiments impossible.

Once the system runs, it will produce a vast amount of output, so experiments must be set up carefully to avoid information

overload. The so-called induction trap—the lure of running too many experiments for each scenario because it is easy to produce more output, and then jumping to inductive conclusions too swiftly [13]—must also be avoided. The output and logs of a set of runs typically do not lend themselves to straightforward reading, but require post-processing. In practice, this means turning huge text files into calculable spreadsheets, and further into graphs and diagrams. Those outputs can then be presented back to the PM, who can then call for more experiments, sensitivity analyses, or even a revision of the requirements specification. The CE in this process makes certain design choices, e.g., which output data to present and how. It is important that this process is iterative and that the PM is given the option of making informed choices, by having at least some grasp of what is realistic to do, given the constraints of computational complexity. The CE must provide technical specifications on further experiments, and the technical competence used also comes with a responsibility to inform: the PM must know what options there are, and why and how certain results were omitted or deemed irrelevant. Because the PM is typically the one responsible for acting upon results obtained, a chain of trust to the CE must be upheld. Likewise, the CE should react if the PM, for example, calls only for certain experiments to be run, or if the selection is made so as to confirm a preconceived truth, in a pseudo-scientific fashion [14].

In principle, the output of the executable model can finally be validated by comparing its predictions to real outcomes of actual policy interventions for the population modeled, given that the input parameters adequately model the real population prior to those interventions. Naturally, some scenarios could be considered extreme (e.g., the introduction of an entirely new influenza virus to a population without native immunity) and are simulated precisely because they cannot be studied in the real world. In such scenarios, validation can, at best, pertain only to parts of the model. More importantly, simulations of outbreaks are difficult to validate because the simulated event is rare. Catastrophic events are characterized by low probability and disastrous consequences (see, e.g., [15]), and yet the input data are collected from the normal state of the population in non-outbreak situations. Using this input, the simulator is expected to produce one possible yet highly unlikely scenario to provide researchers and policy makers with more opportunities to observe and learn about the unlikely event.

Since computational epidemiology is problem-oriented and constitutes applied science, models are often pragmatic in the sense that they are adapted to their use as policy-supporting tools. Any provisos made have to be grounded in the culture of the decision making entity, such as a government or a pharmaceutical company, making alignment studies, in which models are docked for replication studies [16] difficult.

2.2. The Micro-Meso-Macro Link

In microsimulation models of outbreaks, individuals are exposed to the disease and may infect other individuals that they come into contact with. The most primitive unit is the individual and the focus is on the activities of the individual, for the purposes of studying transmission. By contrast, macrosimulation focuses not on the individual, but on the whole society. All members (i.e., the whole, possibly stratified, population) share the same properties and

move between different disease states such as susceptible, infected, and resistant.

Even if originally conceived as a pure microsimulation model, the executable model discussed here has macro-level parameters, e.g., workplace size. This parameter governs how many colleagues a working individual interacts with during a working day. To "interact with" here means that there is an opportunity for infection, given that either the individual or the colleague is ill. Even though micro data are available for each workplace—including the number of employees at each company—it is defensible not to use these data in full, since large workplaces have so many employees that it makes no sense to assume that the individual interacts with them all. In reality, the individual might not even see more than a fraction of the total number of colleagues on a given day. The workplace size is therefore set to a precise value, meant to capture an average number of colleagues, which is kept constant throughout a set of runs.

By definition, macro models do not represent local interaction. However, in any dynamic model utilizing micro data, including SIR-inspired individual-based models [17], local interaction will affect the output. If there appear discernible patterns in the output that are not explicitly stated by the model description at the outset, they are referred to as emergent patterns. In the described model, all output logs are mapped onto a real population. This means that every discernible pattern has an interpretation that can be understood in the epidemiological context, using terms such as "spread" and "giant component", and also in the societal context, using terms like "number of infected" and "absenteeism". Hence, patterns discernible at the macro level resulting from local interactions at the micro level are easily made understandable to the PM.

The meso layer [18] includes everything that is more general than the properties of single individuals but less general than the properties of the whole society. In the model at hand, this is most visible in neighborhoods, defined by the geographical proximity of different households. Adding the meso layer to an epidemiological model enables researchers to represent a crucial part of human interaction: social contacts outside the home or workplace. This includes encountering others while shopping, and social gatherings of neighbors.

Variables in the executable model represent properties of the real population, but many of them cannot be observed directly. Therefore, the argument goes, a suitable value for the executable must be determined by experimenting with the simulator. In the implementation phase, the CE strives to get a handle on the parameter space, i.e. the value space for all parameters that can be subject to variation. To illustrate this, a sample case is now considered.

To find a suitable value for the parameter *context contacts*, representing the average number of contacts outside the home or the workplace, the behavior of the simulated outbreak is observed using the total number of infected individuals per week for a large interval of context contact values. The interval is set to start from zero, where the model behavior is undefined, to where the parameter no longer has an observable impact, i.e. when it is high enough to exhaust the population regardless of all other parameters. Within the [8,20] interval, changing the *context contacts* parameter had, in this example, a significant effect on the behavior of the model. Repeating the same series of experiments with a smaller step size within the [8,20] interval, a smaller region

of interest was obtained within the [14,16] interval. Finally, the analysis was repeated one last time for the [14,16] interval with a smaller step size. Figure 2 shows the number of infections per week for five runs where all parameters except *context contacts* were kept constant. Further simulations were run to observe the effects of variation due to random seeds when contacts was set to 15. Figure 3 shows the number of infections per week for three runs with different random seeds where all other parameters were kept constant.

Other variables in the executable model that should be decided using a similar process include (but are not limited to): number of initially infected, office size, place choice based on disease level, place choice based on age, length of a work day, and the probability of receiving a symptomatic disease profile.

2.3. Stochasticity

An outbreak of pandemic influenza is a rare event. To trigger such an outbreak, either the simulations must be run repeatedly for a long period until an outbreak occurs, or the model must be configured in such a way that outbreaks will occur with higher frequency than in the real world. The former is not practical since it might take millions of runs before anything happens, and the latter comes with the risk of compromising the validity of output by introducing exogeneous variables that change the effects of the simulated outbreak.

All random events in the model use a series of numbers that are generated at runtime using the initial seeds provided by the user. Therefore, the outcome of every "random" event in a simulation run depends only on the initial seeds. By using the same seeds, identical results can be obtained using different computers, operating systems, or compilers.

In the present model, one highly influential parameter is the number of initially infected. When 50 randomly selected individuals are infected, an outbreak is triggered in nearly every run. If only three individuals are selected instead, the outbreaks become much more rare. This is due to the heterogeneity of the population: individuals with more contacts are more likely to initiate outbreaks if infected, and it is more likely that a highly connected individual would be infected if 50 rather than three are infected initially.

It is often assumed in executable models that in a few generations, a simulation with three infected would reach the stage with 50 infected, and that the difference between them would be negligible. Certainly every simulation with three initially infected would reach a stage with 50 infected, given that an outbreak occurs during the run. Therefore simulations can be started from the stage where 50 individuals are infected since that is the minimum number at which the simulation platform produces outbreaks in the majority of runs. This assumption is far from ideal. The simplest observable effect is that no runs will have less than 50 infected. This is acceptable because the object of study is nationwide outbreaks. However, the difference between the two approaches is not negligible because 50 randomly selected individuals will not have the same geographical distribution as 50 individuals whose infections originate from three individuals. The 50-from-three group will most likely have overlapping social networks because they were all infected by three individuals, as opposed to being randomly selected from a popula-

tion of nine million. As the outbreak grows to one thousand or one hundred thousand infected, the difference may lose its significance, but quantifying that significance remains challenging for all executable models that use heterogeneous populations. Hence, this is a good example of a simulation in which the CE makes an assumption about things beyond the PM's control, or even grasp. Good software development requires that such assumptions be made explicit and communicated to the PM.

3. Conclusion

The lessons learned from the software development project described above can be summarized in the form of a check list. Even if the list is not exhaustive, developers of computational epidemiology models could check off the items on the list, as applicable to their project. The presented workflow and checklist do not include surveillance in computational epidemiology and instead focus on modeling and simulation. A more comprehensive workflow for computational epidemiology would have to incorporate computer-assisted infectious disease surveillance, often performed using complex software platforms tailored to the task [19, 20, 21, 22], and the interaction of its users with the actors already identified in the preceding sections.

Computational epidemiology is a new area, and many of the methods and theories employed have yet to benefit from thorough scientific investigation. Even if important steps towards amalgamating models and performing alignment experiments have been taken (see, e.g., [23]), the area is in need of extensive methodological advancement. The following checklist is intended to be a contribution to such development. Not every item in the check list introduces new issues for policy makers or computational epidemiologists, but, depending on the reader's area of expertise, one or two are highly likely to be more significant than the others. Much of it is part of the folklore of the area, and could be classified as procedural and pragmatic know-how. More specifically, the contribution is to have these items made explicit as one concise list, and tied to working procedures as demonstrated by our workflow description (Figure 1).

1. All population data sets are regional

To have access to data on the entire population on the planet is not a realistic goal. Hence, most studies are limited to one geographic region, such as a city, a state, or a country [24]. This means that the universe of discourse includes not only the individuals in this geographic region, but also that a certain proportion of the individuals must be allowed to leave the region. Moreover, visitors and immigrants from other regions should be included in the population data. Some computational epidemiology projects employing micro data use census data, others extrapolate from samples, and yet others use synthetic data. In the rare cases where registry data is available for a large population—as is the case for the Swedish population—hard methodological questions must still be answered regarding the generalizability of results: which parts of a scenario execution in Sweden are likely to be analogous to ones in Norway, Iceland, or the state of Oregon?

2. Population data are sensitive

Even after extensive post-processing, any data set with real population data is subject to privacy and integrity concerns. In almost all countries, this means that running a simulator with the data set is subject to applying to an ethics board. If approved, data

must be kept safe and experiments may be run in designated facilities only. This makes replication studies difficult, and it also restricts alignment studies to less interesting data sets.

3. Verifying the simulator is a serious engineering challenge

To formally verify that the simulator produces adequate results, is free from programming bugs, and can handle the computational complexity of modeling large outbreaks is, in general, not possible. The software is too large, as is the variation of possible input values and the spectrum of sensitivity analyses. Extensive testing—varying the hardware environment and the parameter values, including the random seeds for stochastic processes—yields evidence for adequacy, but no guarantees. This does not entail that the simulator is without use, or not to be trusted, but merely that its construction and maintenance is an engineering challenge.

4. Validating the simulator output is hard

Pandemics have been few and far between. Modeling a future scenario on a real outbreak of the past has been done with some success in the area of epidemiology. The structural properties of current and future societies may vary greatly from those studied in the past, however. Air travel, hygiene, and working conditions are three out of many factors that affect the spread of communicable disease and that vary greatly in the historical perspective. The low probability of catastrophic events such as a pandemic makes it very hard to validate any simulation experiment against real-world events.

5. Assumptions and hypotheses should be stated and controlled by the policy maker

Placing assumptions on top of assumptions will only create a gap between the policy maker and the computational epidemiologist. As illustrated by the example of selecting different initially infected individuals, the description of a single assumption can be interpreted in multiple ways, and the implementation of different interpretations can diverge significantly from the respective intention. The complexity of communicating all assumptions implied by the decisions of the policy maker arises from the tremendous difficulty in identifying implicit assumptions at every step of development. Because every addition to the model carries the risk of modifying the interaction of existing parameters, ensuring that all assumptions have been made by the policy maker becomes a formidable challenge.

6. Triggering outbreaks in the simulator is nontrivial

To implement a simulator that always produces outbreaks is easy. Increasing the infectiousness of a disease (as done, e.g., [17]) or the number of initially infected, quickly yields a disease pattern affecting the entire giant component, i.e. every individual connected to other individuals through the social network or by geographical proximity (cf. [25]), forming the largest connected subgraph of the population graph (cf. [26]). If such settings are inconsistent with empirical data, or with assumptions and hypotheses declared, however, then the adequacy of the model should be questioned. There is evidence for the fact that the initial stages of a pandemic require a different kind of modeling than the later stages [27]. It would therefore be naïve to think that increasing the number of initially infected—in order to trigger outbreaks in a larger proportion of runs—would not affect the model of the entire pandemic.

7. Hybrid models need constant refinement

A model in which the micro, meso, and macro properties are integrated has the potential to mirror reality in a relatively accurate way. Under the proviso that model adequacy yields better prediction, one could discard the simplest models in favour of such hybrid models. The level of ambition, however, comes at the price of the model never being finished, and model-dependent artifacts becoming more difficult to identify. Since the world to be modeled is a moving target, and since macro data can often be replaced by micro data as it becomes available, there are always refinements to be made. The devil is in the details.

Acknowledgements

The authors would like to thank the current leader of the MicroSim project at the Swedish Institute for Communicable Disease Control, Lisa Brouwers. The authors also thank Olof Görnerup, Eric-Oluf Svee, the editor, and the anonymous reviewers for their constructive comments.

References

[1] R. Ross, An Application of the Theory of Probabilities to the Study of a priori Pathometry. Part I, Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 92 (1916) 204–230.

[2] W. O. Kermack, A. G. McKendrick, A contribution to the mathematical theory of epidemics, Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 115 (1927) 700–721.

[3] M. Boman, E. Holm, Multi-agent systems, time geography, and microsimulations, in: M.-O. Olsson, G. Sjöstedt (Eds.), Systems Approaches and their Application, Springer, 2004, pp. 95–118.

[4] S. Eubank, H. Guclu, V. Kumar, M. Marathe, A. Srinivasan, N. Toroczkai Zand Wang, Modelling disease outbreaks in realistic urban social networks, Nature 429 (2004) 180–184.

[5] N. M. Ferguson, D. A. T. Cummings, S. Cauchemez, C. Fraser, S. Riley, A. Meeyai, S. Iamsirithaworn, D. S. Burke, Strategies for containing an emerging influenza pandemic in Southeast Asia, Nature 437 (2005) 209–214.

[6] I. M. Longini, A. Nizam, S. Xu, K. Ungchusak, W. Hanshaoworakul, D. A. T. Cummings, M. E. Halloran, Containing pandemic influenza at the source, Science (New York, N.Y.) 309 (2005) 1083–1087.

[7] D. Balcan, B. Goncalves, H. Hu, J. J. Ramasco, V. Colizza, A. Vespignani, Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model, Journal of Computational Science 1 (2010) 132 – 145.

[8] J. M. Epstein, Agent-based computational models and generative social science, Complexity 4 (1999) 41–60.

[9] Swedish Institute for Communicable Disease Control, Microsim source code, `https://smisvn.smi.se/sim/`, 2010.

[10] L. Brouwers, M. Boman, M. Camitz, K. Mäkilä, A. Tegnell, Micro-simulation of a smallpox outbreak using official register data, Eurosurveillance 15 (2010).

[11] F. Coelho, O. Cruz, C. Codeco, Epigrass: A tool to study disease spread in complex networks, Source Code for Biology and Medicine 3 (2008).

[12] Statistics Sweden, `http://www.scb.se`, 2010.

[13] K. Popper, Philosophy of science: A personal report, British philosophy in mid-century (1957) 182–83.

[14] I. Lakatos, Science and pseudoscience, in: Philosophical Papers Vol. 1, Cambridge University Press, 1977, pp. 1–7.

[15] R. Thom, Structural stability and morphogenesis: An outline of a general theory of models, Addison-Wesley, 1993.

[16] R. Axtell, R. Axelrod, J. M. Epstein, M. D. Cohen, Aligning simulation models: A case study and results, Computational & Mathematical Organization Theory 1 (1996) 123–141.

[17] B. Roche, J.-F. Guegan, F. Bousquet, Multi-agent systems in epidemiology: a first step for

computational biology in the study of vector-borne disease transmission, BMC Bioinformatics 9 (2008).
[18] H. Liljenström, U. Svedin (Eds.), Micro, meso, macro: Addressing complex systems couplings, World Scientific, 2005.
[19] J. Espino, M. Wagner, C. Szczepaniak, F. Tsui, H. Su, R. Olszewski, Z. Liu, W. Chapman, X. Zeng, L. Ma, Z. Lu, J. Dara, Removing a barrier to computer-based outbreak and disease surveillance – The RODS Open Source Project, MMWR Morb Mortal Wkly Rep. 53 Supplement (2004) 32–39.
[20] M. Crubezy, M. O'Connor, Z. Pincus, M. Musen, D. Buckeridge, Ontology-centered syndromic surveillance for bioterrorism, Intelligent Systems, IEEE 20 (2005) 26–35.
[21] D. Abramson, B. Bethwaite, C. Enticott, S. Garic, T. Peachey, A. Michailova, S. Amirriazi, Embedding optimization in computational science workflows, Journal of Computational Science 1 (2010) 41 – 47.
[22] B. Cakici, K. Hebing, G. M., P. Saretok, A. Hulth, CASE: A framework for computer supported outbreak detection, BMC Med Inform Decis Mak 10 (2010).
[23] M. E. Halloran, N. M. Ferguson, S. Eubank, I. M. Longini, D. A. T. Cummings, B. Lewis, S. Xu, C. Fraser, A. Vullikanti, T. C. Germann, D. Wagener, R. Beckman, K. Kadau, C. Barrett, C. A. Macken, D. S. Burke, P. Cooley, Modeling targeted layered containment of an influenza pandemic in the United States, PNAS 105 (2008) 4639–4644.
[24] D. L. Chao, M. E. Halloran, V. J. Obenchain, I. M. Longini, Jr, FluTE, a Publicly Available Stochastic Influenza Epidemic Simulation Model, PLoS Comput Biol 6 (2010) e1000656.
[25] M. Youssef, R. Kooij, C. Scoglio, Viral conductance: Quantifying the robustness of networks with respect to spread of epidemics, Journal of Computational Science In Press, Accepted Manuscript (2011) –.
[26] M. E. J. Newman, The structure and function of complex networks, SIAM Review 45 (2003) 167–256.
[27] E. Bonabeau, L. Toubiana, A. Flahault, The geographical spread of influenza, Proceedings of the Royal Society B 265 (1998) 2421–2425.

Figures

Figure 1 - Executable model workflow

The schematic workflow of developing and running an executable model, incorporating policy makers and computational epidemiologists.

Figure 2 - Context contacts variation

Number of infections per week for five runs where all parameters except context contacts were kept constant.

Figure 3 - Random seed variation

Number of infections per week for three runs with different random seeds where all other parameters were kept constant. Each random seed is a vector of numbers generated by a pseudo-random number generator.

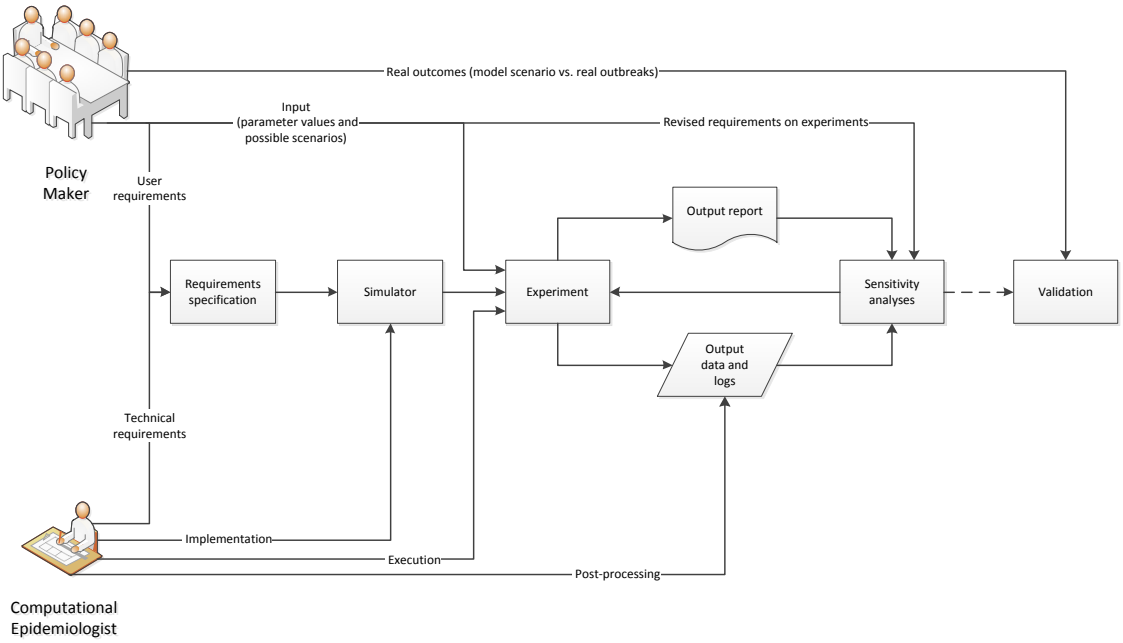


Figure 1: Executable model workflow

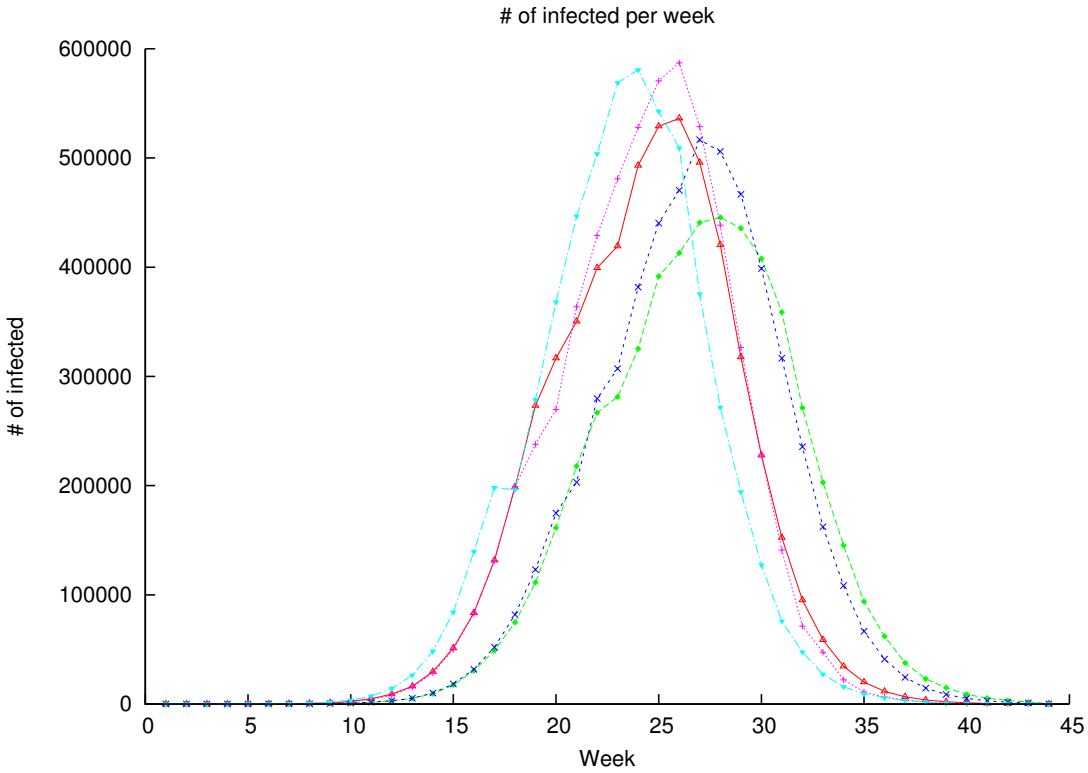


Figure 2: Context contacts variation

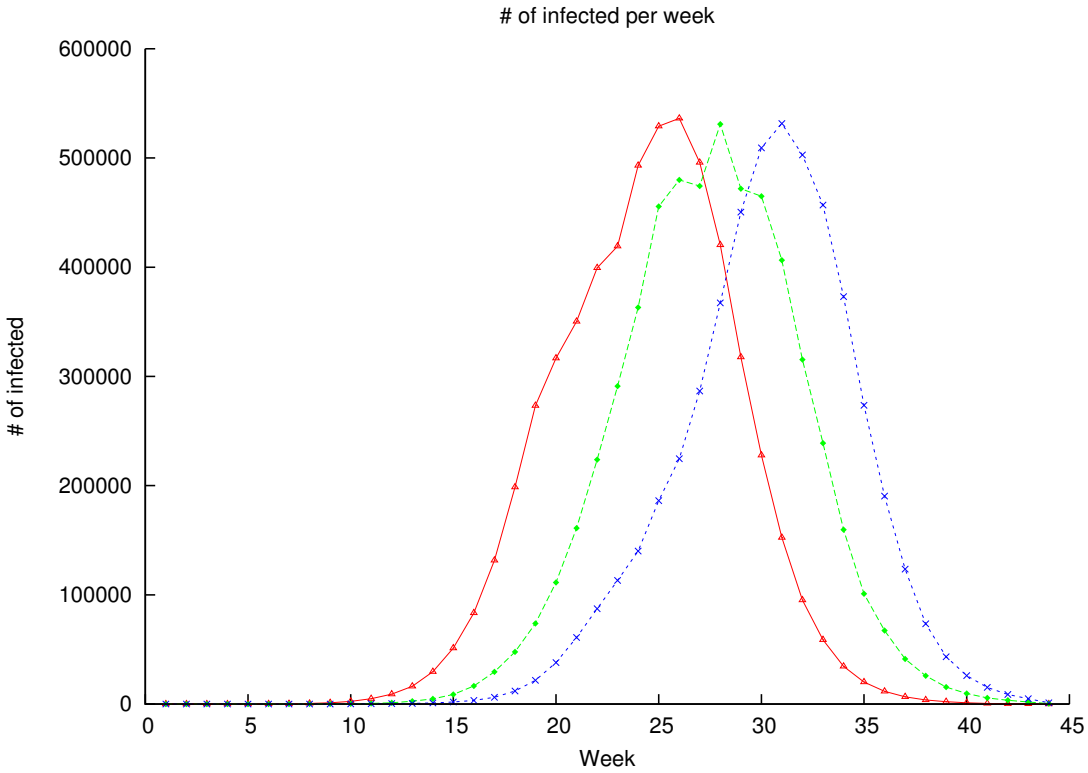


Figure 3: Random seed variation

References

Babin, S., Magruder, S., Hakre, S., Coberly, J. & Lombardo, J. S. (2007), Understanding the data: Health indicators in disease surveillance, *in* J. S. Lombardo & D. L. Buckeridge, eds, 'Disease Surveillance: A Public Health Informatics Approach', first edn, Wiley-Interscience, chapter 2, pp. 43–90.

Berger, M., Shiau, R. & Weintraub, J. M. (2006), 'Review of syndromic surveillance: implications for waterborne disease detection', *Journal of Epidemiology and Community Health* **60**(6), 543–550.

BioSTORM (2009), 'Welcome to BioSTORM', `http://biostorm.stanford.edu/doku.php`. Accessed 2011-04-06.

BioSTORM (2010), 'BioSTORM SVN Repository', `https://bmir-gforge.stanford.edu/gf/project/biostorm/scmsvn`. Accessed 2011-04-06.

Boscoe, F. P., McLaughlin, C., Schymura, M. J. & Kielb, C. L. (2003), 'Visualization of the spatial scan statistic using nested circles', *Health & Place* **9**(3), 273–277.

Bradley, C. A., Rolka, H., Walker, D. & Loonsk, J. (2005), 'BioSense: implementation of a national early event detection and situational awareness system.', *MMWR. Morbidity and Mortality Weekly Report* **54**, 11.

Bravata, D. M., McDonald, K. M., Smith, W. M., Rydzak, C., Szeto, H., Buckeridge, D. L., Haberland, C. & Owens, D. K. (2004), 'Systematic review: Surveillance systems for early detection of Bioterrorism-Related diseases', *Annals of Internal Medicine* **140**(11), 910–922.

Brillman, J. C., Joyce, E. L., Forslund, D. W., Picard, R. R., Umland, E., Koster, F., Sailor, W. C., Judd, S. L., Froman, P., Kellie, S., Kesler, D., Nolte, K. B., Geoge, J. E., Bersell, K., Castle, S. & Albanese, B. (2003), 'The biosurveillance analysis, feedback, evaluation, and response (B-SAFER) system', **80**(S1), i119–i120.

Brookmeyer, R. & Stroup, D. F. (2003), *Monitoring the Health of Populations: Statistical Principles and Methods for Public Health Surveillance*, 1 edn, Oxford University Press, USA.

Brownstein, J. S., Freifeld, C. C., Reis, B. Y. & Mandl, K. D. (2008), 'Surveillance sans frontières: Internet-based emerging infectious disease intelligence and the healthmap project', *PLoS Med* **5**(7), e151.

Buckeridge, D. L., Burkom, H., Campbell, M., Hogan, W. R. & Moore, A. W. (2005), 'Algorithms for rapid outbreak detection: a research synthesis', *Journal of Biomedical Informatics* **38**(2), 99–113.

Buckeridge, D. L., Graham, J., O'Connor, M. J., Choy, M. K., Tu, S. W. & Musen, M. A. (2002), 'Knowledge-based bioterrorism surveillance.', *Proceedings of the AMIA Symposium* pp. 76–80.

Buckeridge, D. L., Musen, M. A., Switzer, P. & Crubézy, M. (2003), A modular framework for automated Space-Time surveillance analysis of public health data, *in* 'AMIA Annual Symposium, Washington, DC', pp. 120–124.

Buckeridge, D. L., Okhmatovskaia, A., Tu, S., O'Connor, M., Nyulas, C. & Musen, M. A. (2008), 'Understanding detection performance in public health surveillance: Modeling aberrancy-detection algorithms', *Journal of the American Medical Informatics Association* **15**(6), 760–769.

Buehler, J. W., Hopkins, R. S., Overhage, J. M., Sosin, D. M. & Tong, V. (2004), 'Framework for evaluating public health surveillance systems for early detection of outbreaks', *MMWR Morbidity and Mortality Weekly Report* **53**, 1–11.

Buehler, J. W., Sonricker, A., Paladini, M., Soper, P. & Mostashari, F. (2008), 'Syndromic surveillance practice in the united states: findings from a survey of state, territorial, and selected local health departments', *Advances in Disease Surveillance* **6**(3), 1–20.

Burkom, H. (2007), Alerting algorithms for biosurveillance, *in* J. S. Lombardo & D. L. Buckeridge, eds, 'Disease Surveillance: A Public Health Informatics Approach', first edn, Wiley-Interscience, chapter 4.

Cakici, B., Hebing, K., Grunewald, M., Saretok, P. & Hulth, A. (2010), 'CASE: a framework for computer supported outbreak detection', *BMC Medical Informatics and Decision Making* **10**, 14.

CDC (2008), Biosense technical overview of data collection, analysis, and reporting, Technical report, Centers for Disease Control and Prevention.

CDC (2011), 'CDC BioSense Home', `http://www.cdc.gov/biosense`. Accessed 2011-04-01.

Chen, H., Zeng, D. & Yan, P. (2009), *Infectious Disease Informatics: Syndromic Surveillance for Public Health and Bio-Defense*, first edn, Springer.

Chung, K., Yang, D. & Bell, R. (2004), 'Health and GIS: toward spatial statistical analyses', *Journal of Medical Systems* **28**(4), 349–360.

Clarke, K. C., McLafferty, S. L. & Tempalski, B. J. (1996), 'On epidemiology and geographic information systems: a review and discussion of future directions.', *Emerging Infectious Diseases* **2**(2), 85–92.

Cohen, H. W., Gould, R. M. & Sidel, V. W. (2004), 'The pitfalls of bioterrorism preparedness: the anthrax and smallpox experiences', *American Journal of Public Health* **94**(10), 1667–1671.

Cooper, D. L., Verlander, N. Q., Smith, G. E., Charlett, A., Gerard, E., Willocks, L. & O'Brien, S. (2006), 'Can syndromic surveillance data detect local outbreaks of communicable disease? a model using a historical cryptosporidiosis outbreak', *Epidemiology and Infection* **134**(1), 13–20.

Cromley, R. G. & Cromley, E. K. (2009), 'Choropleth map legend design for visualizing community health disparities', *International Journal of Health Geographics* **8**(1).

Crubézy, M., O'Connor, M., Buckeridge, D. L., Pincus, Z. & Musen, M. A. (2005), 'Ontology-Centered syndromic surveillance for bioterrorism', *IEEE Intelligent Systems* **20**(5), 26–35.

Diamond, C. C., Mostashari, F. & Shirky, C. (2009), 'Collecting and sharing data for population health: A new paradigm', *Health Affairs* **28**(2), 454–466.

Dowling, K. C. & Lipton, R. I. (2005), 'Bioterrorism preparedness expenditures may compromise public health', *American Journal of Public Health* **95**(10).

ECDC (2010), 'The European surveillance system (TESSy)', `http://www.ecdc.europa.eu/en/activities/surveillance/Pages/Surveillance_Tessy.aspx`. Accessed 2011-04-19.

Espino, J. U., Wagner, M., Szczepaniak, C., Tsui, F., Su, H., Olszewski, R., Liu, Z., Chapman, W., Zeng, X., Ma, L., Lu, Z. & Dara, J. (2004), 'Removing a barrier to computer-based outbreak and disease surveillance—The RODS open source project', *Syndromic Surveillance* p. 32.

European Commission (2008), 'INFTRANS – infectious diseases: modelling for control', `http://ec.europa.eu/research/fp6/ssp/inftrans_en.htm`. Accessed 2011-04-27.

Fearnley, L. (2008), 'Signals come and go: syndromic surveillance and styles of biosecurity', *Environment and Planning A* **40**(7), 1615–1632.

Fearnley, L. (2010), 'Epidemic intelligence. langmuir and the birth of disease surveillance', *Behemoth: a Journal on Civilisation* **3**.

Freifeld, C. C., Chunara, R., Mekaru, S. R., Chan, E. H., Kass-Hout, T., Ayala Iacucci, A. & Brownstein, J. S. (2010), 'Participatory epidemiology: Use of mobile phones for community-based health reporting', *PLoS Med* **7**(12).

Freifeld, C. C., Mandl, K. D., Reis, B. Y. & Brownstein, J. S. (2008), 'HealthMap: global infectious disease monitoring through automated classification and visualization of internet media reports', *Journal of the American Medical Informatics Association* **15**(2), 150–157.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S. & Brilliant, L. (2009), 'Detecting influenza epidemics using search engine query data', *Nature* **457**(7232), 1012–1014.

Google (2011), 'Google earth', `http://earth.google.com`. Accessed 2011-04-22.

Guglielmetti, P., Coulombier, D., Thinus, G., Loock, F. V. & Schreck, S. (2006), 'The early warning and response system for communicable diseases in the EU: an overview from 1999 to 2005', *Eurosurveillance* **11**(12).

Hayes, M. H. (1996), *Statistical digital signal processing and modeling*, John Wiley & Sons.

HealthMap (2011*a*), 'Global health, local information', `http://healthmap.org/en`. Accessed 2011-04-07.

HealthMap (2011*b*), 'Outbreaks near me', `http://healthmap.org/outbreaksnearme`. Accessed 2011-04-08.

Heffernan, R. et al. (2004*a*), 'New york city syndromic surveillance systems', *MMWR. Morbidity and mortality weekly report* **53**, 25–27.

Heffernan, R. et al. (2004*b*), 'Syndromic surveillance in public health practice, new york city', *Emerg Infect Dis* **10**(5), 858–864.

Henning, K. J. (2004), 'What is syndromic surveillance?', *MMWR. Morbidity and Mortality Weekly Report* **53 Suppl**, 5–11.

Hulth, A., Andrews, N., Ethelberg, S., Dreesman, J., Faensen, D., van Pelt, W. & Schnitzler, J. (2010), 'Practical usage of computer-supported outbreak detection in five european countries', *Eurosurveillance, to appear* **15**(36).

Hulth, A., Rydevik, G. & Linde, A. (2009), 'Web queries as a source for syndromic surveillance', *PloS One* **4**(2).

Hutwagner, L., Thompson, W., Seeman, G. M. & Treadwell, T. (2003), 'The bioterrorism preparedness and response early aberration reporting system (EARS)', *Journal of Urban Health: Bulletin of the New York Academy of Medicine* **80**, i89–i96.

ISDS (2010), Final Recommendation: Core Processes and EHR Requirements for Public Health Syndromic Surveillance, Technical report, International Society for Disease Surveillance.
URL: *http://www.syndromic.org/projects/meaningful-use*

Josseran, L., Nicolau, J., Caillère, N., Astagneau, P. & Brücker, G. (2006), 'Syndromic surveillance based on emergency department activity and crude mortality: two examples', *Eurosurveillance* **11**(12), 225–229.

Kass-Hout, T. A. (2009*a*), 'BioSense: Going forward', `http://www.cdc.gov/biosense/files/BioSense_ISDS.ppt`. Accessed 2011-04-19.

Kass-Hout, T. A. (2009*b*), 'BioSense: next generation', `http://www.cdc.gov/biosense/files/PHIN_PowerPoint_BioSense_Next_Generation_FINAL_508.pdf`. Accessed 2011-04-19.

Kleinman, K., Lazarus, R. & Platt, R. (2004), 'A generalized linear mixed models approach for detecting incident clusters of disease in small areas, with an application to biological terrorism', *American Journal of Epidemiology* **159**(3), 217 –224.

Kleinman, K. P. & Abrams, A. M. (2006), 'Assessing surveillance using sensitivity, specificity and timeliness', *Statistical methods in medical research* **15**(5), 445.

Kulldorff, M., Mostashari, F., Duczmal, L., Yih, W. K., Kleinman, K. & Platt, R. (2007), 'Multivariate scan statistics for disease surveillance', *Statistics in Medicine* **26**(8), 1824–1833.

Lampos, V., Bie, T. & Cristianini, N. (2010), Flu detector - tracking epidemics on twitter, *in* J. L. Balcázar, F. Bonchi, A. Gionis & M. Sebag, eds, 'Machine Learning and Knowledge Discovery in Databases', Vol. 6323, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 599–602.

Lawson, A. B. & Kleinman, K. (2005), *Spatial and Syndromic Surveillance for Public Health*, Wiley.

Leitenberg, M. (2005), *Assessing the Biological Weapons and Bioterrorism Threat*, U.S. Army War College Strategic Studies Institute.
URL: *http://www.strategicstudiesinstitute.army.mil/pubs/display.cfm?pubID=639*

Lombardo, J., Burkom, H., Elbert, E., Magruder, S., Lewis, S. H., Loschen, W., Sari, J., Sniegoski, C., Wojcik, R. & Pavlin, J. (2003), 'A systems overview of the electronic surveillance system for the early notification of community-based epidemics (ESSENCE II)', *Journal of Urban Health: Bulletin of the New York Academy of Medicine* **80**(2 Suppl 1), i32–42.

M'ikantha, N. M., Southwell, B. & Lautenbach, E. (2003), 'Automated laboratory reporting of infectious diseases in a climate of bioterrorism', *Emerging Infectious Diseases* **9**(9), 1053–1057.

Mostashari, F. (2003), 'Syndromic surveillance: a local perspective', *Journal of Urban Health: Bulletin of the New York Academy of Medicine* **80**, i1–i7.

Naumova, E. N., O'Neil, E. & MacNeill, I. (2005), 'INFERNO: a system for early outbreak detection and signature forecasting', *MMWR. Morbidity and Mortality Weekly Report* **54 Suppl**, 77–83.

NNDSS (2010), `http://www.health.gov.au/internet/main/Publishing.nsf/Content/cda-surveil-nndss-nndssintro.htm`. Accessed 2011-04-19.

Nykiforuk, C. I. J. & Flaman, L. M. (2011), 'Geographic information systems (GIS) for health promotion and public health: A review', *Health Promotion Practice* **12**(1), 63–73.

O'Connor, M. J., Buckeridge, D. L., Choy, M., Crubézy, M., Pincus, Z. & Musen, M. A. (2003), 'BioSTORM: a system for automated surveillance of diverse data sources', **2003**, 1071–1071.

Pelecanos, A., Ryan, P. & Gatton, M. (2010), 'Outbreak detection algorithms for seasonal disease data: a case study using ross river virus disease', *BMC Medical Informatics and Decision Making* **10**(1), 74.

Pincus, Z. & Musen, M. A. (2003), Contextualizing heterogeneous data for integration and inference, *in* 'AMIA 2003 Symposium Proceedings', pp. 514–518.

Reingold, A. (2003), 'If syndromic surveillance is the answer, what is the question?', *Biosecurity and Bioterrorism: Biodefense Strategy, Practice, and Science* **1**(2), 77–81.

Reis, B. Y., Kirby, C., Hadden, L. E., Olson, K., McMurry, A. J., Daniel, J. B. & Mandl, K. D. (2007), 'AEGIS: a robust and scalable real-time public health surveillance system', *Journal of the American Medical Informatics Association: JAMIA* **14**(5), 581–588.

RODS (2009), 'The RODS Open Source Project', `http://openrods.sourceforge.net`. Accessed 2011-04-04.

Rolfhamre, P., Jansson, A., Arneborn, M. & Ekdahl, K. (2006), 'SmiNet-2: description of an internet-based surveillance system for communicable diseases in sweden', *Eurosurveillance* **11**(5), 103–107.

Sidel, V., Gould, R. & Cohen, H. (2002), 'Bioterrorism preparedness: cooptation of public health?', *Medicine & Global Survival* **7**(2), 82–89.

Sonesson, C. & Bock, D. (2003), 'A review and discussion of prospective statistical surveillance in public health', *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **166**(1), 5–21.

Sosin, D. M. & DeThomasis, J. (2004), 'Evaluation challenges for syndromic surveillance – Making incremental progress', *MMWR. Morbidity and mortality weekly report* **53**, 125.

Tolentino, H., Kamadjeu, R., Fontelo, P., Liu, F., Matters, M., Pollack, M. & Madoff, L. (2007), 'Scanning the Emerging Infectious Diseases Horizon-Visualizing ProMED Emails Using EpiSPIDER', *Advances in Disease Surveillance* **2**, 169.

Travers, D., Barnett, C., Ising, A. & Waller, A. (2006), Timeliness of emergency department diagnoses for syndromic surveillance, *in* 'AMIA Annual Symposium Proceedings', Vol. 2006, p. 769.

Tsui, F., Espino, J. U., Dato, V. M., Gesteland, P. H., Hutman, J. & Wagner, M. M. (2003), 'Technical description of RODS: a real-time public health surveillance system', *Journal of the American Medical Informatics Association: JAMIA* **10**(5), 399–408.

van den Wijngaard, C., van Pelt, W., Nagelkerke, N., Kretzschmar, M. & Koopmans, M. (2011), 'Evaluation of syndromic surveillance in the netherlands: its added value and recommendations for implementation.', *Eurosurveillance* **16**(9).

Wagner, M. M., Moore, A. W. & Aryel, R. M. (2006), *Handbook of Biosurveillance*, first edn, Academic Press.

Wagner, M. M., Robinson, J., Tsui, F., Espino, J. U. & Hogan, W. R. (2003), 'Design of a national retail data monitor for public health surveillance', *Journal of the American Medical Informatics Association* **10**(5), 409–418.

Wong, W. & Moore, A. (2006), Classical time-series methods for biosurveillance, *in* M. M. Wagner, A. W. Moore & R. M. Aryel, eds, 'Handbook of biosurveillance', Academic Press, chapter 14.

Yih, W. K., Caldwell, B., Harmon, R., Kleinman, K., Lazarus, R., Nelson, A., Nordin, J., Rehm, B., Richter, B., Ritzwoller, D., Sherwood, E. &

Platt, R. (2004), 'National bioterrorism syndromic surveillance demonstration program', *MMWR. Morbidity and Mortality Weekly Report* **53 Suppl**, 43–49.

Zelicoff, A., Brillman, J., Forslund, D. W., George, J. E., Zink, S., Koenig, S., Staab, T., Simpson, G., Umland, E. & Bersell, K. (2001), The rapid syndrome validation project (RSVP), *in* 'Proceedings of the AMIA Symposium', pp. 771–775.

Zeng, D., Chen, H., Tseng, C., Larson, C., Chang, W., Eidson, M., Gotham, I., Lynch, C. & Ascher, M. (2005), BioPortal: sharing and analyzing infectious disease information, *in* P. Kantor, G. Muresan, F. Roberts, D. D. Zeng, F. Wang, H. Chen & R. C. Merkle, eds, 'Intelligence and Security Informatics', Vol. 3495, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 612–613.

Zhang, J., Tsui, F. C., Wagner, M. M. & Hogan, W. R. (2003), 'Detection of outbreaks from time series data using wavelet transform', *AMIA Annual Symposium Proceedings* pp. 748–752.