Simulating and Evaluating Biosurveillance Datasets

Thomas H. Lotze
Applied Mathematics and Scientific Computation Program
University of Maryland
College Park, Maryland

Galit Shmueli
Department of Decision, Operations & Information Technologies and Center for Health and Information Decision Systems
University of Maryland
College Park, Maryland

Inbal Yahav
Robert H. Smith School of Business
University of Maryland
College Park, Maryland

CONTENTS
2.1 Motivation ................................................................. 24
2.2 Data Simulation .......................................................... 26
  2.2.1 Overview .............................................................. 26
  2.2.2 Creating Initial Multivariate Data ....................... 26
  2.2.3 Adding Effects of Holidays, Seasonality, and DOW  28
    2.2.3.1 Holidays ......................................................... 28
    2.2.3.2 Seasonality .................................................... 28
    2.2.3.3 Day-of-Week (DOW) ..................................... 28
2.3 Mimicking Existing Dataset Qualities .......................... 28
2.4 Evaluating Simulation Effectiveness ............................. 30
  2.4.1 Univariate $\chi^2$ Tests ..................................... 30
  2.4.2 Multivariate Tests .............................................. 31
2.5 Outbreak Signature Simulation ................................... 32
  2.5.1 Overview .......................................................... 32
  2.5.2 Outbreak Signature Types ................................. 34
2.1 Motivation

The field of biosurveillance involves the monitoring measures of diagnostic and prediagnostic activity for the purpose of finding early indications of disease outbreaks. By providing early notification of potential outbreaks, the aim is to provide public health officials the opportunity to respond earlier and thus more effectively. Although the field has grown in importance and emphasis in the past several years, the research community involved in designing and evaluating monitoring algorithms has not grown as expected. A major barrier has been data accessibility: typically researchers do not have access to biosurveillance data unless they are part of a biosurveillance group. In fact, after parting from a biosurveillance group, researchers lose their data access. This means that a very limited community of academic researchers works in the field, with a nearly impenetrable barrier to entering it (especially for statisticians or other nonmedical academics). Furthermore, the confinement of each research group to a single source of data and the lack of data sharing across groups “leaves opportunity for scientific confounding” (Rolka, 2006).

While simulated data have their own difficulties, they seem to be a necessity for modern biosurveillance research. Buckeridge et al. (2005) explain:

[They] are appealing for algorithm evaluation because they allow exact specification of the outbreak signal, perfect knowledge of the outbreak onset, and evaluators can create large amounts of test data …
The International Society for Disease Surveillance (ISDS) has recognized the paucity of data (both authentic and realistic simulations) as an issue that hinders the field’s progression, and is currently working to create a data repository for publicly available datasets. It recently sponsored a contest using simulated data, but required all participants to delete the data after the completion of the contest. There is a serious need for datasets, which simulation holds promise for alleviating.

The first implementation of wholly simulated biosurveillance data in the form of daily counts is the publicly available simulated background and outbreak datasets by Hutwagner et al. (2005). The background series are generated from a negative-binomial distribution with parameters set such that “means and standard deviations were based on observed values from national and local public health systems and biosurveillance surveillance systems. Adjustments were made for days of the week, holidays, postholiday periods, seasonality, and trend.” Other research, such as Fricker et al. (2008), has simulated background data using an additive combination of terms representing level, seasonal and day-of-week effects, and random noise. Our approach, as described in Section 2.2, is similar in that we set or estimate levels and temporal patterns from authentic data. However, our approach is more general in that it captures two key dependence structures: autocorrelation and cross-correlation. In particular, we include 1-day autocorrelation, which has been shown to be a major property of biosurveillance daily time series (Burkom et al., 2007) and generate multivariate rather than univariate data: we generate a set of time series rather than a single time series at a time. Thus, there can be a dependence structure between these series (in the form of cross correlations).

More recently, Siddiqi et al. (2007) developed a simulation method based on linear dynamical systems, also known as Kalman filters. They model the observed series as a linear transformation from a series of latent variables, find a stable linear transformation for those latent variables, and use this transformation to re-create similar data and to extend it into the future. They modify standard Kalman filter methods, incrementally adding constraints to create a system whose linear transformation remains stable (with eigenvalues less than 1). This method seems very promising, and we recommend using the methods described here to evaluate its effectiveness at mimicking authentic data.

Finally, we note that to evaluate an algorithm’s performance on biosurveillance data, one must be able to simulate outbreak signals within the data. It is common practice to evaluate algorithms by seeding real biosurveillance data with simulated outbreak signals (e.g., Burkom et al., 2007; Goldenberg et al., 2002; Reis and Mandl, 2003; Stoto et al., 2006, and many others). However, simulating these outbreak signals accurately is even more difficult than simulating the background biosurveillance data, as known examples of outbreak signatures in health care seeking behavior are even more difficult to obtain. We emphasize that generating a realistic multivariate outbreak signal must be based on epidemiological and other relevant domain knowledge.
2.2 Data Simulation

2.2.1 Overview

As noted in Buckeridge et al. (2005), the main challenge is “complexity of simulating background and outbreak signal,” and in particular,

To allow for meaningful evaluation of diverse algorithms, both normal and outbreak data must be simulated in a manner that ensures sufficient complexity and validity in terms of factors such as spatial patterns, temporal patterns, and joint distributions of variables. As a simulation model grows to meet these requirements, the number of parameters increases, the ability to verify the model becomes difficult, and ultimately it becomes more difficult to ensure the validity of the simulated data.

Our approach is thus to identify those features that seem central in authentic data, estimate the appropriate parameters from authentic data, and use them to stochastically generate new data. In particular, we use the statistical structure of authentic multivariate time series derived from biosurveillance data in order to simulate background data that have the same structure. We can even mimic a particular dataset, thereby generating one or more stochastic duplications of it.

Our method for simulating multivariate time-series data includes several prominent patterns that have been shown by various empirical studies to exist in biosurveillance time series. Day-of-week (DOW) is a common pattern. In emergency department visits in the United States, daily counts are typically lower on weekends and high during the week (Burkom et al., 2007), but can also exhibit other daily patterns (e.g., Brillman et al., 2005; Reis and Mandl 2003), or none (Fricker 2006). Grocery stores tend to have more traffic on weekends, and therefore medication sales appear higher on weekends (e.g., Goldenberg et al., 2002). Another common pattern is abnormal behavior on holidays and postholidays (e.g., Fienberg and Shmueli, 2005; Zhang et al., 2003) due to holiday closings (e.g., schools) or limited operation mode (e.g., pharmacies, hospitals). Another pattern exhibited by some series is seasonal cyclical behavior such as annual or biannual (summer/winter) fluctuations. The daily frequency of collection also leads to nonnegligible short-term autocorrelation (see, e.g., Burkom et al., 2007; Lotze et al., 2008). Finally, there are also dependencies between series that manifest as cross correlations.

Our simulator begins by generating “simple” multivariate time series that include autocorrelation and cross correlation, and then add to them DOW, seasonal, and holiday effects.

2.2.2 Creating Initial Multivariate Data

We generate a set of initial multivariate data from a multivariate normal distribution in the following way: a vector of means, a vector of variances, and a
correlation matrix (or equivalently, a vector of means and a covariance matrix) are provided by the researcher. In addition, a few optional parameters can be specified: (1) a vector of autocorrelations to induce autocorrelations into each series; (2) a random seed for ensuring repeatability of generation; and (3) the length of the series to be generated.

A covariance matrix is created from the variance vector and the correlation matrix. If there is no autocorrelation, the covariance matrix is used to generate a series of independent multivariate normal random data. In the presence of non-zero autocorrelation, the covariance matrix is used to generate the first day of data, and each subsequent day of data are then generated from the conditional multivariate normal distribution given counts on the previous day. This maintains the same covariance overall but also includes autocorrelation. Specifically, we represent the vector of values on \( k \) series at day \( t \) as \( \bar{X}_t = \left[ X_{1,t} \quad \ldots \quad X_{k,t} \right] \), with mean \( \bar{\mu} = \left[ \mu_1 \quad \ldots \quad \mu_k \right] \) and covariance matrix \( \Sigma \). The bivariate distribution of \( \bar{X}_{t+1} \) and \( \bar{X}_t \) is

\[
\begin{pmatrix}
X_{1,t+1} \\
X_{2,t+1} \\
\vdots \\
X_{k,t+1}
\end{pmatrix}
| \bar{X}_t \sim N\left( \left( \bar{\mu}, \Sigma \right) \right),
\]

where \( C \) is a diagonal matrix with elements \( c_i (i = 1, \ldots, k) \) on the diagonal, where \( c_i = Cov(X_i, t, X_i, t+1) \) is the lag-1 autocovariance of series \( i \). Then, given the values on day \( t \), the conditional distribution of the next day (with the given covariance \( \Sigma \) and autocovariance \( C \)) is

\[
\bar{X}_{t+1} \mid \bar{X}_t \sim N\left( \bar{\mu}^*, \Sigma^* \right),
\]

where \( \bar{\mu}^* = \bar{\mu} + \Sigma^{-1}(\bar{X}_t - \bar{\mu}) \) and \( \Sigma^* = \Sigma - \Sigma^{-1}C\Sigma \).

Data generated from this conditional distribution provide a multivariate dataset with the given means, covariance, and autocorrelation structure.

The next step is to add effects such as DOW and seasonality to the initial data. To do that, we first “label” the initial data by creating indicators for DOW, day, month, and year. Now each day of data has a calendar date attached to it.
2.2.3 Adding Effects of Holidays, Seasonality, and DOW

The inclusion of the three types of patterns to the initial data is done sequentially. We describe a certain order, but if all components are entered either multiplicatively or additively (rather than a mix), then the order of pattern inclusion does not matter.

2.2.3.1 Holidays

Holiday effects are added either in multiplicative form (the new point is a fraction of the original) or in additive form (the new point is the original, with some amount subtracted). Holidays can be specified at any point in the series; the default is to use multiplicative holiday effects on all federal holidays (derived from the office of personnel management site, www.opm.gov/fedhol: New Year’s Day, birthday of Martin Luther King, Jr., Washington’s Birthday, Memorial Day, Independence Day, Labor Day, Columbus Day, Veterans Day, Thanksgiving Day, and Christmas Day).

2.2.3.2 Seasonality

Seasonality is added in either additive or multiplicative form. It can be a scalar (in which case, it modifies a shifted sine wave function with a period of 365.25 days, \( f = \sin(2\pi * x/(365.25) + 2) \)) or a fully specified vector. The default is to add no seasonality; a multiplicative one-half-scale sine wave appears to be a good approximation for respiratory seasonality.

2.2.3.3 Day-of-Week (DOW)

The DOW pattern can be multiplicative or additive, and must be fully specified as a vector containing an index for each day. By default, it is set to multiplicative, with weekends set to one-third of weekday values.

In each of these steps, the dataset is normalized to maintain the same means (by dividing by or adding the appropriate amount to the series overall). However, the covariance does increase as the effects are applied uniformly to each series.

Finally, after all patterns of interest are added, the series are rounded to integers and bounded to be nonnegative in order to yield valid count data.

2.3 Mimicking Existing Dataset Qualities

In addition to generating a general type of multivariate time series with specific temporal and dependence patterns, our data simulator can also be used
to mimic a multivariate authentic dataset, thereby producing a new semiauthentic dataset. The authentic dataset and its semiauthentic mimic have the same statistical structure, yet they differ in their actual daily counts. This combination means that the resulting semiauthentic datasets are useful for purposes of research (e.g., algorithm development and evaluation), yet avoid data disclosure concerns such as privacy and confidentiality.

Another important use of simulated datasets that are “copies” of the same authentic dataset is for purposes of randomization and Monte Carlo testing. The ability to test an algorithm on multiple versions of the same data structure helps avoid overfitting and gives more accurate estimates of model performance.

In the following, we describe how the different statistical components of the authentic data are estimated. These are then used to create the mimicked dataset.

Estimating DOW patterns. Given a dataset, the method of ratio to moving averages (RMA) is used to estimate DOW indices. A vector of seven indices is created separately for each series, in order to capture the weekly pattern for that series.

Estimating seasonality. The data are smoothed using a 7-day moving average. A smoothing spline is fit to the smoothed data, and this spline is then evaluated at each daily point. These daily points are then used as the seasonality components. For more details on the smoothing spline, see Chambers and Hastie (1992).

Estimating holiday effects. Holiday dates are copied from the original dataset, if any are present. This vector is used to identify that days should have holiday effects in the mimicked dataset.

Estimating series means. The mean for each simulated series is determined from the mean of the corresponding original series, excluding holidays (if any).

Estimating series variances and autocorrelations. To determine the variance and autocorrelation of the authentic series devoid of seasonal patterns, we use a Holt–Winters exponential smoother on each series separately and obtain a series of residuals (actual daily counts minus predicted daily counts). This residual series should not contain trends, DOW, or seasonal effects. We then compute the autocorrelation, variance, and correlation matrix of the residual series, which is later used as input for the simulator.

Some or all of the previous estimated parameters are then fed into the data simulator, thereby yielding a simulated mimicked version of the authentic multivariate data. The original dataset and its mimic contain the same statistical characteristics but are different stochastic realizations (i.e., the counts are not identical).
2.4 Evaluating Simulation Effectiveness

A crucial component of using simulation to mimic authentic data is verifying that the simulated data retain the key characteristics of the original data. This is done by testing whether the simulated data come from the same distribution as the original authentic data. If they come from the same distribution, then the simulation method should be trustworthy and provide valid results; if not, then the differences between the original and simulated data can provide distorted and unrealistic results.

Of course, given a finite amount of original data, there exist an infinite number of distributions that could generate that data. The distribution tests used here merely attempt to confirm that the simulation method is within that space of possible models, specifically those that have a reasonable chance of generating the data. We must use domain knowledge (such as our awareness of which characteristics are relevant) to further constrain the possible simulation models. Goodness-of-fit tests of the simulated data should be considered as relative measures of consistency; it is known that distributional tests become extremely sensitive with large amounts of data, and so may reject even the most useful simulations.

In addition, it should be considered that a mimic method will only be useful if it accurately captures the randomness of the underlying distribution. If a mimic is simply a duplicate of the original data, it is clearly not a good additional test, nor does it avoid any privacy concerns. Similarly, a mimic that merely adds random noise to the original is not providing a new authentic set of possible data; it is simply providing the original data with extra variation.

2.4.1 Univariate $\chi^2$ Tests

The first method for evaluating the closeness between distribution of authentic and mimic data is a series of simple $\chi^2$ tests. To test a mimic against its original dataset, we take each univariate data series and split it by day of week. The values for a single day of week are then formed into bins; an example of the binning process is given in Figure 2.1. The width of the bin varies by density, determined such that there are at least 10 observations in each bin. The original data are split and binned in the same fashion, and these two sets of counts (mimicked and original) are tested for distributional equality using a $\chi^2$ test (with degrees of freedom equal to $k - 1$, where $k$ = the number of bins). An FDR (Benjamini and Hochberg 1995) significance correction is used to account for multiple testing across multiple series. The $\chi^2$ tests can also be repeated for each DOW separately with FDR correction, not only to inform us whether there are issues with our simulation but also to point us toward the reasons for those issues.
2.4.2 Multivariate Tests

The preceding $\chi^2$ tests can only uncover univariate disparities between the original and mimicked data. To also consider the covariance between the series, we consider multivariate goodness-of-fit tests. While it is not obvious that such a test can be performed in a distribution-free manner, several methods have been developed to do so (notably, Bickel 1969; Friedman and Raisky 1979; Schilling 1986; Kim and Foutz 1987; Henze 1988; Hall and Tajvidi 2002).

In this chapter, we use the nearest-neighbors test described in Schilling (1986), because of its asymptotic normality and computational tractability. Under this test, the nearest $k$ neighbors are computed for the combined sample. Each of the nearest neighbors is then used to determine an indicator variable, whether or not it shares the same class as the neighboring point. The statistic $T$, the proportion of $k$-nearest neighbors sharing the same class, is used to test equality of distributions. If both samples have the same size and come from the same distribution, $T$ will approach 0.5 as the sample size increases. If the two samples differ in distribution, then $T$ will tend to be larger than 0.5. With an appropriate correction, $T$ has an approximate standard normal distribution. For an example, see Figure 2.2.

![Figure 2.1](image-url)

**Figure 2.1**
A portion of a single time series being binned.
2.5 Outbreak Signature Simulation

2.5.1 Overview

We also consider the simulation of outbreak signatures to be added to the background data. In order to compare the performance of biosurveillance algorithms in terms of true and false alert rates and timeliness, we simulate not only background data but also outbreaks signatures that can be injected into the background data. Because in biosurveillance, the nature of the outbreak signatures is generally unspecified, algorithms are tested across different types and sizes of outbreak signatures.

Most researchers evaluating algorithm performance have added simulated outbreak signatures to authentic data. Many studies add a fixed number of additional cases, a linearly growing number of cases, or an exponentially growing number of cases to the authentic data (Goldenberg et al., 2002; Reis et al., 2003; Reis and Mandl 2003; Mandl et al., 2004; Stoto et al., 2006) in order to provide a variety of different possible outbreak signal shapes. Single-day “spike” signals and multiday lognormal curves are also popular (Burkom 2003b; Burkom et al., 2007), as they have some epidemiological basis. Spike
outbreak signals occur when the disease onset and spread is faster than the rate of reporting for the health sources, or when the disease effect is rapid and tightly peaked. Lognormal outbreak signals are more common: many diseases are seen to have a lognormal distribution after time of infection to when symptoms develop and are reported (incubation time).

Buckeridge et al. (2004) has proposed a realistic model that includes modeling of anthrax patterns using a plume model for dispersion as well as modeling incubation period and behavior of the infected population. Wallstrom et al. (2005) present a software tool (HiFide) for modeling cryptosporidium outbreak signals and influenza in univariate health series. Both of these models are based on real outbreaks. STEM (Ford et al., 2006) is a software plug-in that can be used to quickly generate outbreak signals with a variety of infection parameters, over real geographic transportation networks. Other methods, which simulate individual cases (Wong et al., 2002) or spatiotemporal data (Cassa et al., 2005; Watkins et al., 2007), can be adapted to generate daily counts.

However, we again caution that such outbreak signal simulations can currently only be judged via domain knowledge; there is not enough data to compare their accuracy using statistical tests. For this reason, our outbreak simulator extends outbreak simulation to the multivariate case. Because an outbreak will likely manifest in multiple related series, we must be able to simulate an outbreak signal which occurs in each. The simulator can generate both spike and lognormal shapes, in a variety of sizes (increased number of affected cases, possibly different for different series) and shapes; it allows flexibility to tailor the outbreak generation as appropriate for the comparison.

In addition, we provide a novel labeling system that takes the multivariate nature of the data into account. There is still debate as to what time period of an outbreak it is valuable to detect. For example, if a detected alert occurs after the peak of the disease effect, it is not very useful to the public health practitioners. In general, for univariate series, the debate is between counting any alert during an outbreak versus counting only alerts that occur before the peak of the disease outbreak signal. For the multivariate case, this is even more complicated, as the peaks may occur at different times in different series. For this reason, instead of only two labels (normal/outbreak), we use four labels as follows:

- **0**—no outbreak signal on that day
- **a**—outbreak before any series have peaked
- **b**—outbreak between the first and the last series’ peak
- **c**—outbreak after all series have peaked

Note that labels are applied to days, not to single series. An example is shown in Figure 2.3, where a single lognormal outbreak signal was placed on day 10 and injected into both series. Until day 9, the label is 0. On day 10, an
outbreak signal starts, and hence the label is $a$. The outbreak signal reaches its peak in series 2 on day 12 and in series 1 on day 14. The outbreak label for days 12–14 is therefore $b$. Finally, on day 15, the number of cases decreases, and the corresponding outbreak label is $c$.

Finally, we also allow for an outbreak to be generated and then modified according to the same effects (DOW, holiday, seasonal, etc.) as the simulated background data. Such a modification is possible with simulated background health data, as the parameters are unknown in actual data. Although it is unknown whether outbreak signals are subject to the same effects as the background health data, a reasonable assumption is that the same reasons that keep people from showing up in no-outbreak scenarios (weekend, weather, etc.) will affect them equally when they are sick due to a “normal” cause or a “disease outbreak” cause.

### 2.5.2 Outbreak Signature Types

The outbreak signature simulation enables the user to generate two types of signatures: a single-day (multivariate) spike and a multiple-day (multivariate) log-normal progression. As in the data simulator, we start with an “initial” outbreak signature, and then add to it patterns such as seasonality, DOW, and holidays.

To create the initial outbreak signature, one must set for each series $n_{outbreak}$, the increase in the total number of cases throughout the outbreak manifestation period (which can be thought of as the total number of cases added due to the outbreak). Users can either manually define a vector of additional cases for each series, or they can specify the increase in the series mean in terms of a multiple of the standard deviation.
Algorithm 1 Create log-normal outbreak

**Input:** $[\mu, \sigma, \text{numCases}]$

- samples = Generate $\text{numCases}$ samples from a log-normal distribution with mean = $\mu$ and sd = $\sigma$
- outbreak = histogram(sample)
- trim $t\%$ last cases from outbreak

### 2.5.2.1 Single-Day Spike

Generating a single-day spike outbreak requires either specifying $n_{\text{outbreak}}$ directly for each series, or it can be set as a multiple of the standard deviation of the initial series:

$$n_{\text{outbreak}} = \text{const} \times \text{std}$$

We anticipate that some users will be more comfortable with specifying the total count increase directly, while others will prefer to determine the increase in terms of standard deviations (as is customary in statistical process control). In general, we consider small-to-medium spike sizes, because biosurveillance systems are designed to detect early and more subtle indications of a disease outbreak.

### 2.5.2.2 Lognormal Outbreak Signature

A lognormal progression is a reasonable epicurve model, because as Burkom (2003a) describes, the incubation period distribution of many infectious diseases tends to be lognormal, with distribution parameters dependent on the disease agent and route of infection. Generating a lognormal signature such as the spike requires specifying the size of the outbreak $n_{\text{outbreak}}$. The main difference is that this quantity now spans over more than a single day such that $n_{\text{outbreak}}$ is in fact the area under the lognormal curve. Similar to the spike, $n_{\text{outbreak}}$ can either be set directly or as a multiple of the standard deviation of the series (excluding DOW, seasonal, and holiday effects).

In addition to the signature size, the user must specify its shape, determined by the lognormal distribution parameters $\mu$ and $\sigma$. The peak of the outbreak signature (which corresponds to the mode of a lognormal distribution) is approximately on day $\exp(\mu - \sigma^2)$. We then trim the latest $t\%$ of the cases to avoid long tails. This process is summarized by algorithm 1, and an example of a simulated lognormal outbreak is given in Figure 2.4.

### 2.5.2.3 Adding Effects to Initial Outbreak Signatures

To this “initial” outbreak, effects such as DOW, seasonal patterns, and holidays can be added in the same way that they are added to the initial simulated data (see Section 2.3).
2.6 Example: Mimicking a BioALIRT Dataset

2.6.1 Mimicking Background Health Data

To illustrate the product of the data simulator, we mimic a set of authentic multivariate daily counts taken from the BioALIRT program conducted by the U.S. Defense Advanced Research Projects Agency (DARPA) (Siegrist and Pavlin 2004). We use a dataset of six series from a single city, where three of the series are indicators of respiratory symptoms and the other three of gastrointestinal symptoms. The series come from three different data sources: military clinic visit diagnoses, filled military prescriptions, and civilian physician office visits, all within a particular U.S. city. Figure 2.5 displays the six series of daily counts over a period of nearly 2 years.

To mimic this six series dataset, we estimate the different explainable patterns using the mimicker, and then generate a mimicked copy of the original dataset. Figure 2.6 displays the six mimicked series. This dataset is clearly very similar to the authentic dataset in its overall appearance in terms of count levels and patterns. To better see the similarity in structure with respect to the cyclic behavior (seasonality in Resp and DOW in both Resp and GI),

![Figure 2.4](image_url)

**FIGURE 2.4**
Log-normal outbreak with 1000 cases. Outbreak shape: $\mu - 2 \sigma - 0.5$. 
Figure 2.5: Authentic data: Daily counts of respiratory- and gastrointestinal-related doctors’ visits (military and prescription) and filled prescriptions for a particular given US city.
FIGURE 2.6
Mimicked daily counts of the authentic data in Figure 2.5.
Figures 2.7, 2.8, and 2.9 show the pairs of authentic (black) and mimicked (gray) series on different temporal scales: daily, weekly, and monthly. Although the authentic dataset and mimicked dataset appear very similar, they are far from identical. By generating the mimicked dataset stochastically, we obtain a different realization from the same process. To see how the daily counts differ between the authentic and mimicked data, see Figure 2.10, which displays the differences between the daily counts of each authentic series and its mimicked counterpart. We see that the differences are in the order of magnitude of tens of counts. There are also several days with extreme deviations between the authentic and mimicked series. These are mostly on days that are either non-federal holidays (e.g., Christmas Eve and New Year’s Eve) or federal holidays on which “business is as usual” in many areas (e.g., Columbus Day). This emphasizes the importance of specifying all relevant holidays in the particular area where the data are collected or simulated.

2.6.2 Distribution Testing

We now consider the tests of distributional equivalence. The multivariate nearest-neighbor test gives a raw statistic of 0.536, which after standardization provides a Z-score of 3.62, with a p-value of 0.000293. These p-values should be viewed cautiously, because due to the sample size of \( n = 1400 \), it will be very sensitive to any differences in distribution. Comparing it to another earlier simulation method using different DOW variances shows improvement, compared to the alternative method’s standardized Z-score of 33.3. Still, the value is quite low, leading us to consider the univariate \( \chi^2 \) tests.

When the individual DOW scores are considered for each series, we find significant deviations in four categories: giMilVisit on Sun (p-val = 0.000915); giMilVisit on Sat (p-val = 0.000225); giPrescrip on Sun (p-val = 0.000045); and giCivVisit on Sun (p-val = 0.000060).

Examining individual bin comparisons, we see that the mimics have less variance on weekends than the original, suggesting that a negative binomial with increased variance might improve the simulation method. Figure 2.11 shows differences in Sundays for GI Civilian visits.

Outbreaks can then be inserted into this simulated dataset, to provide labeled semiauthentic health data. Such data are necessary in order to apply many detection or classification algorithms.

2.6.3 Outbreak Insertion

In the next step, we simulate an outbreak signature and then insert it into the mimicked data. For illustration, we simulated a lognormal outbreak signature with parameters \( \mu = 0, \sigma = 1 \), and \( n_{\text{outbreak}} \sim N(2\sigma, 2) \). Figure 2.12 displays
FIGURE 2.7
Authentic (black) and mimicked (gray) series, displayed at daily frequency. Series are in the same order as in Figures 2.5 and 2.6.
FIGURE 2.8
Authentic (black) and mimicked (grey) series, displayed at weekly frequency. Series are in the same order as in Figures 5 and 6.
FIGURE 2.9
Authentic (black) and mimicked (grey) series, displayed at monthly frequency. Series are in the same order as in Figures 5 and 6.
Simulating and Evaluating Biosurveillance Datasets

Figure 2.10
Daily count deviations between each of the six authentic series and their mimics.
An indication of a difference between authentic and mimicked data: the mimicked series tend to have lower variance than the authentic data.
FIGURE 2.12
Simulated lognormal signature of an outbreak that starts on October 8, 2001 (Monday), after incorporating the explainable patterns.
Figure 2.13
Final labeled data: mimicked data with inserted outbreak. During the outbreak, the line is bold and colored.
the simulated outbreak signature. Before inserting it into the mimicked dataset, we first add the explainable effects to the outbreak data.

Now the simulated outbreak signature (that includes the explainable patterns) is inserted into the mimicked data, and labels are applied to each day according to the labeling scheme described in the previous section. The end result can be seen in Figure 2.13.

This entire process can then be repeated, inserting outbreaks on different days, to create multiple datasets with different outbreak locations. This provides a large number of example datasets with similar background data and outbreak type, but which are stochastically different. Thus, it allows a researcher to run an algorithm many times and summarize the results, estimating an algorithm’s average performance in terms of false alerts and outbreaks detected.

2.7 Summary and Future Work

2.7.1 Future Work

There are several potential improvements that could be made to the mimic methodology. We anticipate that adding more lags to the mimic will increase the accuracy of patterns captured. While most of the autocorrelation is captured by a single day lag, additional lags still hold higher-level information about the series. In addition, a more elaborate spline fitting to estimate seasonal components would be valuable and could potentially allow for extension of mimics to longer series.

An alternative method for simulating health data is to simulate individual-level activities within a city (such as visiting an ED or purchasing medication). This was proposed and implemented in WSARE 3.0 (Wong et al., 2003). These simulated individual-level events could then be aggregated to the level of biosurveillance health series of the type examined here. Alternatively, one could also modify WSARE’s Bayesian method, using the conditional probabilities of case given combinations of characteristics as sufficient statistics from original health data, as another testable way to generate simulated series.

The evaluation tests considered here are unable to detect certain types of deviations between the authentic and mimicked datasets. For example, since the temporal factor is not considered, they will be unable to find differences in autocorrelation and other time-related deviations. For example, if all Saturday values were randomly reordered, the test results would be identical. Similarly, if the daily observations were reordered to have the same marginal distribution, but a different autocorrelation, this ordering would not cause a change in the test results. In addition, these tests will not find cases where the simulated data is too close to the original, such as when there
is simple random variation around the original data points. As described earlier, however, this is an undesirable property of a mimic simulation. Tests for such scenarios should also be considered.

Ultimately, the best test of the mimicked data will be whether algorithms perform equally well on the mimicked data and on authentic data. If detection algorithms perform on authentic data as well as on mimicked data, we can be confident that our mimicked series are useful for testing and comparing algorithms. We can test this by simulating and injecting outbreak signatures, then testing the performance of various algorithms on authentic versus simulated data.

2.7.2 Summary

An R package for the mimic and outbreak functions is freely available at projectmimic.com, along with 10 simulated datasets mimicked from an authentic biosurveillance dataset. The R package is easily installed and contains extensive help for all functions, with example code. The datasets contain 2 years of data, with six health indicators from a single region. We encourage researchers to freely use the code or datasets provided.

By making the code and algorithms public and freely available, we hope to lower the barriers to entry and allow more researchers to become involved in biosurveillance. By providing a mechanism for generating mimics, we hope to encourage data holders to make mimics freely available. By providing a mechanism for testing mimics, we hope to evaluate methods for mimicking time-series data and to improve such methods.

In conclusion, we believe that simulation can be an effective way of generating new, semiauthentic datasets for public research, free from privacy, confidentiality, and proprietary constraints. The tests presented here provide checks on the validity of the simulation, and allow us to consider further improvements in simulation of health data. By doing this, we hope to enable many more researchers to consider the many challenges, in particular statistical, in biosurveillance (see Shmueli and Burkom [2008] for a survey of such challenges) and to provide an opportunity for rapid advancement of both research and practical solutions.

Acknowledgments

We thank Dr. Howard Burkom and Sean Murphy from the Johns Hopkins Applied Physics Laboratory for useful discussion and suggestions. The research was partially funded by NIH grant RFA-PH-05-126. Permission to use the data was obtained through data use agreement #189 from TRICARE Management Activity. For the first author, this research was performed under an appointment to the U.S. Department of Homeland Security (DHS)
Scholarship and Fellowship Program, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by Oak Ridge Associated Universities (ORAU) under DOE contract number DE-AC05-06OR23100. All opinions expressed in this paper are the author’s and do not necessarily reflect the policies and views of DHS, DOE, or ORAU/ORISE.

References


