

surveillance: An R package for the monitoring of infectious diseases

Michael Höhle

Received: 11 January 2007 / Revised: 22 June 2007 / Accepted: 26 June 2007 /
Published online: 15 August 2007
© Springer-Verlag 2007

Abstract Public health surveillance of emerging infectious diseases is an essential instrument in the attempt to control and prevent their spread. This paper presents the R package “surveillance”, which contains functionality to visualise routinely collected surveillance data and provides algorithms for the statistical detection of aberrations in such univariate or multivariate time series. For evaluation purposes, the package includes real-world example data and the possibility to generate surveillance data by simulation. To compare algorithms, benchmark numbers like sensitivity, specificity, and detection delay can be computed for a set of time series. Package motivation, use and potential are illustrated through a mixture of surveillance theory, case study and R code snippets.

Keywords Monitoring · Public health surveillance · Time series of counts · Outbreak detection · Univariate and multivariate surveillance

1 Introduction

Public health authorities have, in an attempt to meet the threats of infectious diseases, created comprehensive mechanisms for the routine collection of disease data. The vast amounts of data resulting from this acquisition demands the development of automated algorithms for the detection of abnormalities. This paper considers the setup, where these data, possibly after an initial processing from the surveillance

M. Höhle (✉)
Department of Statistics, University of Munich, Ludwigstr. 33, Munich 80539, Germany
e-mail: hoehle@stat.uni-muenchen.de

M. Höhle
Munich Center of Health Sciences, Munich, Germany

database, result in univariate or multivariate time series of case counts. Monitoring of such time series typically occurs by a combination of heuristic methods and statistical modelling. Prominent examples of surveillance algorithms are the work by [Stroup et al. \(1989\)](#) and [Farrington et al. \(1996\)](#) applied in e.g. the ISIS project in the Netherlands ([Widdowson et al. 2003](#)) or the monitoring of bacterial gastrointestinal in Denmark ([Ethelberg and Mølbak 2007](#)). A comprehensive survey of outbreak detection methods can be found in ([Farrington and Andrews 2003](#); [Sonesson and Bock 2003](#); [Lawson and Kleinman 2005](#)).

The R package `surveillance` offers an implementation of surveillance algorithms for epidemiologists and an infrastructure for developers within R, a free software environment for statistical computing and graphics ([Development Core Team 2006](#)). Statistically trained analysts can use the implemented algorithms to monitor their data and developers of new algorithms can test and compare results with those of standard surveillance methods. To this end, real-world outbreak datasets are included together with mechanisms for simulating surveillance data. With the package at hand, comparisons between algorithms as in [Hutwagner et al. \(2005\)](#) should be easy to conduct.

This paper is organised as follows. Section 2 gives a brief introduction to surveillance data and illustrates how to create new datasets by simulation. These data are then analysed in Sect. 3, which explains and exemplifies the use of univariate surveillance algorithms. Usage of the package for the visualisation and analysis of multivariate surveillance data is covered in Sect. 4. Finally, Sect. 5 provides a discussion and indicates directions of future work.

2 Univariate surveillance data

Denote by $\{y_t ; t = 1, \dots, n\}$ the time series of counts. Because such data are typically collected on a weekly or monthly basis, the alternative notation $\{y_{i:j}\}$ shall also be used, with $j \in \{1, \dots, 52\}$ or $j \in \{1, \dots, 12\}$ being the week or month in year $i \in \{-b, \dots, -1, 0\}$. That way the years are indexed such that the most recent year has index zero. Without loss of generality I shall in the following assume weekly data. For evaluation of the outbreak detection algorithms it is also possible for each week to store—if known—whether there was an outbreak that week. The resulting time series $\{(y_t, x_t) ; t = 1, \dots, n\}$ is in `surveillance` given by an object of class `disProg` (disease progress), which is basically a `list` containing two vectors: the observed number of counts and a Boolean vector `state` indicating whether there was an outbreak that week. A number of time series are contained in the `data` directory, mainly originating from the `SurvStat@RKI` database maintained by the Robert Koch Institute, Germany ([Robert Koch Institute 2004](#)). For example the object `ha` describes the weekly number of adult male hepatitis A cases in the federal state of Berlin during 2001–2006. In the summer of 2006 the health authorities noticed an increased amount of cases compared to the previous years ([Robert Koch Institute 2006](#)). In `surveillance` this time series is stored as `disProg` object, which contains the counts for each of the 12 districts in Berlin,

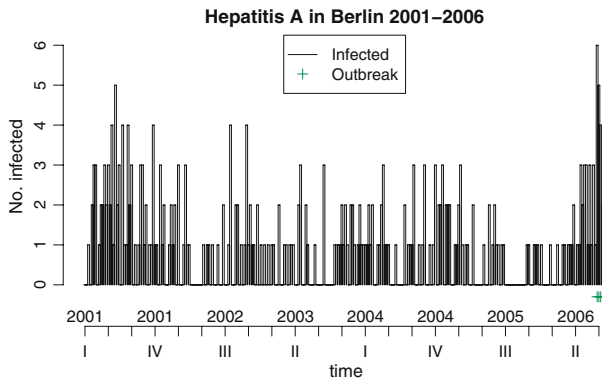


Fig. 1 Weekly adult male hepatitis A cases in Berlin 2001–2006, with the 4-weeks of the known outbreak denoted by ‘+’

but in the following only the time series aggregated over all districts is considered. Visualisation as in Fig. 1 is done as follows.

```
R> data("ha")
R> plot(aggregate(ha), main = "Hepatitis A in Berlin 2001-2006")
```

For test purposes it is also often of interest to generate surveillance data by simulation. A Hidden Markov Model (HMM) is introduced, where a binary state $X_t, t = 1, \dots, n$, denotes whether there was an outbreak and Y_t is the number of observed counts. The state X_t is given by a homogeneous Markov chain with a 2×2 transition matrix specified by two parameters p and r : $P(X_{t+1} = 0|X_t = 0) = p$ and $P(X_{t+1} = 1|X_t = 1) = r$. In addition, the observed Y_t is Poisson-distributed with log-link mean depending on a seasonal effect and time trend, i.e., $\log \mu_t = A \cdot \sin(\omega \cdot (t + \varphi)) + \alpha + \beta t$. In case of an outbreak ($X_t = 1$) the mean increases with a value of K , altogether

$$Y_t \sim \text{Po}(\mu_t + K \cdot X_t). \quad (1)$$

The model in (1) corresponds to a single-source, common-vehicle outbreak, where the length of an outbreak is controlled by the transition probability r and the frequencies of outbreaks by p . The advantage of (1) is that it allows for an easy definition of a correctly identified outbreak: each $X_t = 1$ has to be identified. More advanced setups would require different definitions of an outbreak, e.g., as a connected series of time instances, where the number of outbreak cases is greater than zero. Care is then required in defining what a correctly identified outbreak for time-wise overlapping outbreaks means.

In surveillance the function `sim.pointSource` is used to simulate such a point-source epidemic; the result is an object of class `disProg` as shown in Fig. 2.

```
R> sps <- sim.pointSource(p = 0.99, r = 0.5, length = 400,
+   A = 1, alpha = 1, beta = 0, phi = 0, frequency = 1,
+   state = NULL, K = 1.7)
R> plot(sps, xaxis.years = FALSE)
```

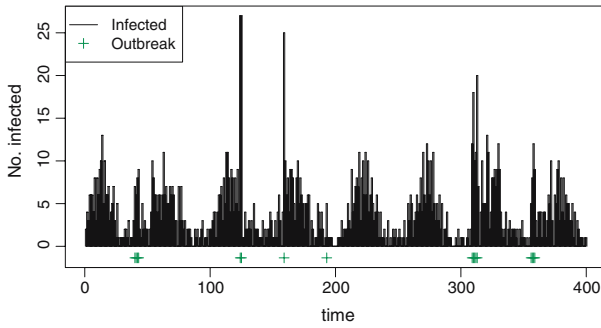


Fig. 2 A simulated time series. The ‘+’ indicate time points, where $X_t = 1$

3 Univariate surveillance algorithms

The aim of surveillance algorithms is the timely detecting of aberrations in surveillance data. *surveillance* offers two classes of algorithms: reference value based and those inspired by statistical process control. The following will exemplify the two classes and discuss how to evaluate their performance.

3.1 Algorithms based on reference values

Surveillance data often exhibit strong seasonality; one way to deal with this fact is through the use of *reference values*: let $y_{0:t}$ be the number of cases of the current week (denoted week t in year 0), b the number of years to go back in time and w the number of weeks around t to include from these previous years. For the year zero we use w_0 as the number of previous weeks to include, typically $w_0 = w$. Altogether the set of reference values is:

$$R(w, w_0, b) = \left(\bigcup_{i=1}^b \bigcup_{j=-w}^w y_{-i:t+j} \right) \cup \left(\bigcup_{k=-w_0}^{-1} y_{0:t+k} \right).$$

This gives the number of cases at time points with similar conditions as at $y_{0:t}$. Note that the number of cases of the current week is not part of $R(w, w_0, b)$.

A surveillance algorithm based on reference values is a procedure using $R(w, w_0, b)$ to create a prediction $\hat{y}_{0:t}$ for the current week. This prediction is then compared with the observed $y_{0:t}$: if the observed number of cases is much higher than the predicted number, the current week is flagged for further investigations. In order to do surveillance for time $0 : t$, an important concern is the choice of b and w . Values as far back as time $-b : t - w$ contribute to $R(w, w_0, b)$ and thus have to exist in the observed time series.

Four different types of algorithms based on reference values are implemented in *surveillance*. The Centers for Disease Control and Prevention (CDC) method (Stroup et al. 1989), the Farrington method (Farrington et al. 1996), the method used

at the Robert Koch Institute (RKI), Germany (Altmann 2003), and a Bayesian approach documented in Riebler (2004). To give an idea of concepts the Bayesian approach is presented.

Here, one assumes independently and identically (iid) Poisson distributed reference values with parameter λ . A gamma distribution is used as prior distribution for λ . The reference values are defined to be $R_{\text{Bayes}} = R(w, w_0, b) = \{y_1, \dots, y_n\}$ and $y_{0:t}$ is the value to predict. Thus, $\lambda \sim \text{Ga}(\alpha, \beta)$ and $y_i | \lambda \sim \text{Po}(\lambda)$, $i = 1, \dots, n$. Standard derivations show that the posterior distribution is

$$\lambda | y_1, \dots, y_n \sim \text{Ga} \left(\alpha + \sum_{i=1}^n y_i, \beta + n \right).$$

Computing the predictive posterior distribution for the next observation

$$f(y_{n+1} | y_1, \dots, y_n) = \int_0^{\infty} f(y_{n+1} | \lambda) f(\lambda | y_1, \dots, y_n) d\lambda,$$

one gets the Poisson-gamma distribution, which is a generalisation of the negative binomial distribution. Altogether,

$$y_{n+1} | y_1, \dots, y_n \sim \text{NegBin} \left(\alpha + \sum_{i=1}^n y_i, \frac{\beta + n}{\beta + n + 1} \right).$$

Using Jeffrey's prior $\text{Ga}(\frac{1}{2}, 0)$ as non-informative prior distribution for λ , the parameters of the negative binomial distribution are

$$\alpha + \sum_{i=1}^n y_i = \frac{1}{2} + \sum_{y_{i:j} \in R_{\text{Bayes}}} y_{i:j} \quad \text{and} \quad \frac{\beta + n}{\beta + n + 1} = \frac{|R_{\text{Bayes}}|}{|R_{\text{Bayes}}| + 1}.$$

Employing a quantile-parameter α , the smallest value y_α is computed, so that $P(y_{n+1} \leq y_\alpha | y_1, \dots, y_n) \geq 1 - \alpha$. Now $A_{0:t} = I(y_{0:t} \geq y_\alpha)$, i.e., if the observed value $y_{0:t}$ is equal to or greater than y_α , then an alarm will be flagged for the current week. For example, the below-stated code applies this procedure with $R(w, w_0, b) = (6, 6, 2)$ and $\alpha = 0.01$ to observations 209–290 from the `ha` dataset, the resulting plot is shown in Fig. 3.

```
R> ha.b662 <- algo.bayes(aggregate(ha), control = list(range = 209:290,
+   b = 2, w = 6, alpha = 0.01))
R> plot(ha.b662, firstweek = 1, startyear = 2005)
```

As an example of applying the more traditional algorithms, Fig. 4 is the result of applying the CDC and Farrington procedure to the simulated time series `sps` from Fig. 2. Note that the CDC procedure operates with 4-week aggregated data—to better compare the upper bound values, the aggregated number of counts for each week are thus shown as circles in the plot.

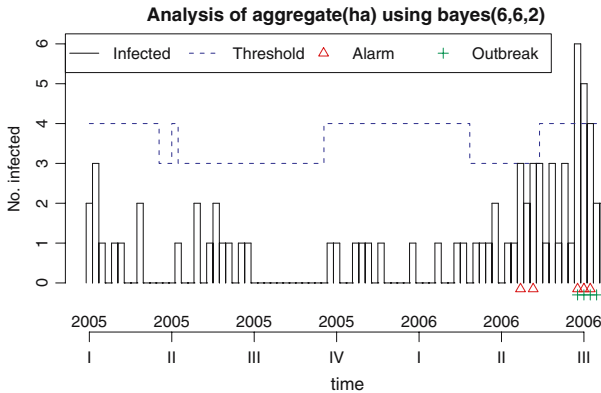


Fig. 3 The Bayes algorithm with $R(w, w_0, b) = (6, 6, 2)$ applied to the ha dataset. Triangles indicate alarms by the algorithm and should be compared to the known outbreaks ('+')

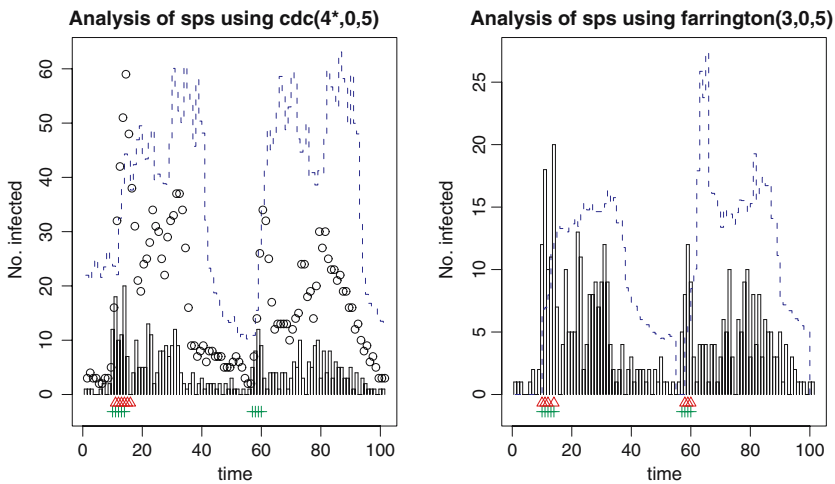


Fig. 4 The CDC (*left*) and the Farrington (*right*) algorithm applied to the simulated time series from Fig. 2

```
R> cntrl <- list(range = 300:400, m = 1, w = 3, b = 5, alpha = 0.01)
R> sps.cdc <- algo.cdc(sps, control = cntrl)
R> sps.farrington <- algo.farrington(sps, control = cntrl)
```

3.2 Algorithms inspired by statistical process control

Because monitoring at each time instance is only based on a subset of the observations, surveillance algorithms based on reference values tend to be sub-optimal. A different class of algorithms is the one inspired by statistical process control (SPC) techniques such as the cumulative sum (CUSUM) chart. In the basic SPC setting one assumes that during an in-control state the observations are iid $x_1, \dots, x_n \sim N(0, 1)$. In case of an out-of-control state, i.e., an outbreak, this distribution changes to $N(\mu, 1)$. Aim

of an SPC chart is now to detect on-line if and when such a change-point occurs. The CUSUM for example accomplishes this by monitoring the time series using the statistic:

$$S_t = \max(0, S_{t-1} + x_t - k), \quad t = 1, \dots, n,$$

where $S_0 = 0$ and k is a user-defined constant called the *reference value* determining the magnitude of change to detect. A change-point is detected once $S_t > h$, where h is called the *decision interval*. An important measure of chart performance is the so called *average run length* (ARL). Typically, two ARLs are reported: the mean time before the first alarm (i) in an in-control state (i.e., no change-point) and (ii) in an out-of-control state (i.e., change-point at time zero). Given desired values for these two ARLs, the function `find.kh` uses the `spc`-package (Knoth 2004) to compute the corresponding h and k .

Several differences between the standard CUSUM chart and charts suited for the surveillance of infectious disease data exist though: the time series consist of count data, which might experience seasonality or other time changing behaviour. To obtain a CUSUM for iid count data, $y_1, \dots, y_n \sim \text{Po}(m)$, Rossi et al. (1999) suggested to transform data to normality by using the following transformation:

$$x_t = \frac{y_t - 3m + 2\sqrt{m \cdot y_t}}{2\sqrt{m}}.$$

In case of time-changing behaviour one loosens the iid assumption through risk-adjusting the chart by letting m be time varying. A suggestion to handle seasonality is to use predictions from a Poisson regression model:

$$\log(m_t) = \alpha + \sum_{s=1}^S \left(\gamma_s \cos(\omega_s t) + \delta_s \sin(\omega_s t) \right),$$

where $\omega_s = \frac{2\pi}{52}s$ are the Fourier frequencies, see Höhle (2006) for details.

The code below performs the above Rossi based seasonal CUSUM detection to the hepatitis data using $S = 1$ with Fig. 5 showing the results. Once the CUSUM signals, no resetting is applied as suggested by Kenett and Pollak (1983), i.e., alarms occur until the S_t statistic again drops below the threshold. CUSUMs are better to detect sustained shifts, which compared to Fig. 3 means that the alarm is sounded earlier.

```
R> kh <- find.kh(ARLa = 500, ARLr = 7)
R> ha.cusum <- algo.cusum(aggregate(ha), control = list(k = kh$k,
+       h = kh$h, m = "glm", trans = "rossi", range = 209:290))
```

The `algo.cusum` function also supports different transformations to normality, e.g., deviance, Pearson or Anscombe residuals (Pierce and Schafer 1986). Simulation studies (not shown) though underline that such detectors are problematic in case of low count numbers, as the normal approximation is not sufficiently warranted here. In these cases the discrepancy between the anticipated and actual ARLs can thus be ample and very instable, see Rogerson and Yamada (2004).

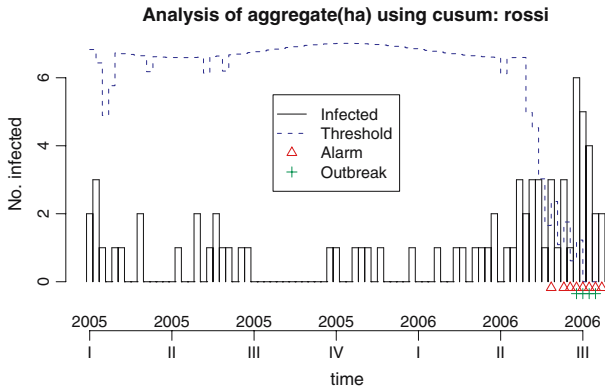


Fig. 5 Rossi CUSUM applied to the hepatitis data using a Poisson regression model with $S = 1$, $\alpha = 0.046$, $\gamma_1 = 0.016$ and $\delta_1 = -0.057$. The upper line shows for each time instance the number of diseased individuals it would have taken the CUSUM to signal

3.3 Algorithm performance

Typically, one is interested in testing and comparing surveillance algorithms. An easy way is to look at the sensitivity and specificity of the procedure. A correct identification of an outbreak is defined as follows: If the algorithm raises an alarm for time t , i.e., $A_t = 1$, and $X_t = 1$, one has a correct classification. If $A_t = 1$ and $X_t = 0$, one has a false positive. In order to compute various performance scores, the function `algo.quality` can be used on a `SurvRes` object.

```
R> print(algo.quality(ha.b662))
TP FP TN FN Sens Spec      dist      mlag
[1,] 3  2  76 1  0.75 0.974359 0.2513115 0
```

This computes the number of false positives, true negatives, false negatives, the sensitivity and the specificity. Finally, `mlag` is the average number of weeks between the first of a consecutive number of $X_t = 1$'s (i.e., an outbreak) and the first alarm raised by the algorithm.

In order to compare the results of different algorithms for a single time series, a list of control objects is declared—each element containing the name and settings of one algorithm to be applied. Testing on a set of time series is performed in the following way. Firstly, a list containing all `disProg` objects is created. Secondly, each algorithm specified in the afore mentioned control object is applied to all series. Consequently, all predefined algorithms based on weekly reference values (saved in `aparv.control`) are applied to the 14 surveillance time series from `SurvStat@RKI` (i.e., the list `outbrks`) by:

```
R> surv.one <- function(outbrk) {
+   algo.compare(algo.call(outbrk, control = aparv.control))
+ }
R> algo.summary(lapply(outbrks, surv.one))
```


	TP	FP	TN	FN	sens	spec	dist	mlag
rki(6,6,0)	38	62	2646	180	0.174	0.977	0.826	5.43
rki(6,6,1)	65	83	2625	153	0.298	0.969	0.703	5.57
rki(4,0,2)	80	106	2602	138	0.367	0.961	0.634	5.43
bayes(6,6,0)	61	206	2502	157	0.280	0.924	0.724	1.71
bayes(6,6,1)	123	968	1740	95	0.564	0.643	0.564	1.36
bayes(4,0,2)	162	920	1788	56	0.743	0.660	0.426	1.36
farrington(6,0,1)	97	140	2568	121	0.445	0.948	0.557	5.07
farrington(4,0,2)	92	126	2582	126	0.422	0.953	0.580	5.43

The above results compare the algorithms at $\alpha = 0.05$ using different amounts of past data. This and previous simulation studies show that the Bayesian approach seems to perform quite well. However, the extent of the above comparisons do not legitimate any more supported statements. Consult the work of [Riebler \(2004\)](#) for a more thorough comparison using simulation studies.

4 Towards multivariate surveillance

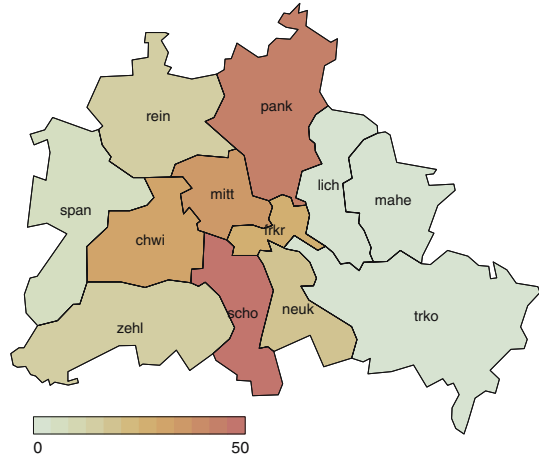
An extension to the surveillance setup described in the preceding sections is the situation where several time series of counts are observed simultaneously. This could e.g., be the same disease observed in multiple regions or the joint observation of several related diseases. The simplest approach to such multivariate surveillance is to simultaneously and independently monitor each region using a univariate method. However, this approach fails to take any correlation between regions or diseases into account and can thus result in poor detection performance. [Sonesson and Frisén \(2005\)](#) provide a recent review of more genuine multivariate surveillance techniques.

Representation and visualisation of multivariate time series data occurs through the already familiar `disProg` and `survRes` S3 classes. However, first steps towards the more formal S4 class system are made by the class `sts` (surveillance time series), which covers the contents of `disProg` and `survRes`. Thus an `sts` object contains matrices with observations, state, population counts, alarm bounds and alarms for each observational unit and time point.

```
R> setClass("sts", representation(week = "numeric", freq = "numeric",
+   start = "numeric", observed = "matrix", state = "matrix",
+   alarm = "matrix", upperbound = "matrix", neighbourhood = "matrix",
+   populationFrac = "matrix", map = "SpatialPolygonsDataFrame",
+   control = "list"))
```

Two additional slots `neighbourhood` and `map` capture the spatial dimension: the former specifies the spatial or structural relation between the units of observation, the latter allows a specification of the spatial structure through a `SpatialPolygonsDataFrame` object from the `sp` package ([Pebesma and Bivand 2005](#)). Using the `mapproj` package ([Lewin-Koh and Bivand 2006](#)), it is easy to use shapefiles from geographical information systems (GIS) for this visualisation. To illustrate, the hepatitis data from Sect. 2 are re-visited by converting them to an `sts` object. Figure 6 shows the time-aggregated counts for each of the twelve districts in Berlin, which makes apparent that most of the cases originate from regions near the city center.

Fig. 6 Spatial visualisation of the time aggregated hepatitis data for each district in Berlin



```
R> shp <- system.file("shapes/berlin.shp", package = "surveillance")
R> ha <- disProg2sts(ha, map = readShapePoly(shp, IDvar = "SNAME"))
R> plot(ha, type = observed ~ 1 | unit)
```

Using `type = observed~1 | time*unit` instead would have created an animation consisting of a picture like Fig. 6 for each time index. The initial visualisation suggests a closer look at the six districts Pankow, Mitte, Friedrichshain-Kreuzberg, Tempelhof-Schöneberg, Charlottenburg-Wilmersdorf and Neukölln. To robustify surveillance, counts for each district are aggregated in four week blocks. Figure 7 shows the result of the following CUSUM surveillance.

```
R> ha4 <- aggregate(ha[, c("pank", "mitt", "frkr", "scho",
+ "chwi", "neuk")], nfreq = 13)
R> ha4.cusum <- cusum(ha4, control = list(k = 1.5, h = 1.75,
+ m = "glm", trans = "rossi", range = 52:73))
R> plot(ha4.cusum, type = observed ~ time | unit)
```

Independent univariate CUSUMs thus sound alarms for Pankow, Mitte and Tempelhof-Schöneberg, but as Fig. 7 shows: detection might have been quicker if dependencies between districts had been utilised. Currently, the package implements only multivariate monitoring by the independent univariate methods from Sect. 3, but work is in progress to implement some of the techniques mentioned in [Sonesson and Frisén \(2005\)](#) and to add detection to the model of [Held et al. \(2005\)](#).

5 Discussion and future work

Surveillance provides a framework for the application of surveillance algorithms using the freely available environment for statistical computing R. Combining the functionality of R with Sweave ([Leisch 2003](#)) and LaTeX allows for easy access to SQL databases, GIS systems and automatic generation of reports. Importing respective R packages to accomplish tasks such as ARL calculation and map visualisation shows how beneficial it can be to reside within a free software environment. The package

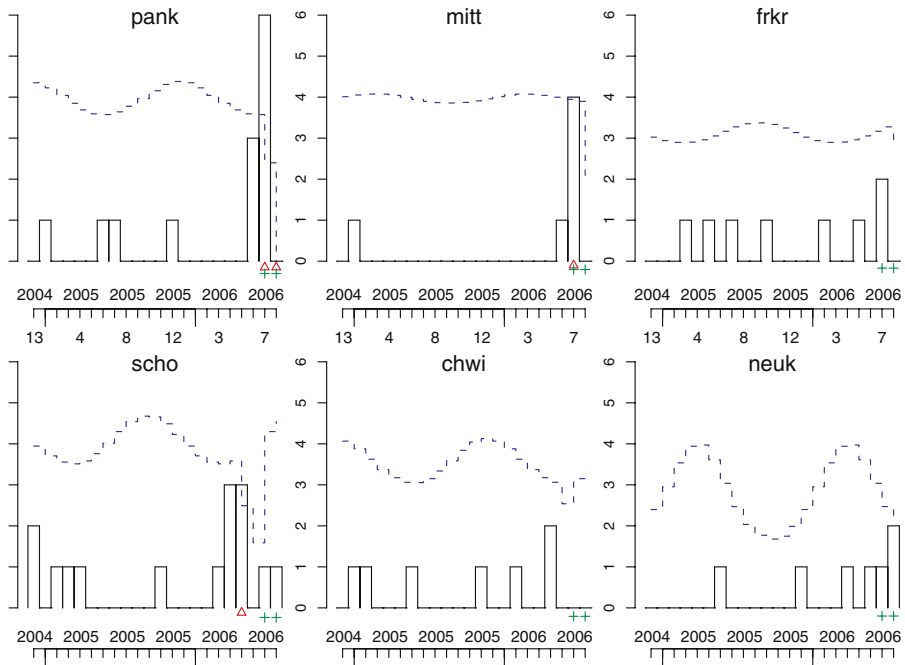


Fig. 7 Hepatitis surveillance results for the six selected districts of Berlin with 4-week aggregation

is available under the GPL licence from the comprehensive R Archive Network at <http://cran.r-project.org>. With `demo("cost")` the analyses in this article can be reproduced.

Casting surveillance algorithms into a Bayesian framework and thus interpreting alarm thresholds as quantiles of the posterior predictive distribution gives a new way to see outbreak detection. Yet an important issue remains multiple testing and the choice of threshold. Here, evaluation based on ARLs, conditional delays or receiver operating characteristic (ROC) curves should be preferred.

Several extensions of the algorithms described in Sect. 3 are conceivable: clever ways to handle reporting delay, an estimation procedure correcting for past outbreaks and treating the inherent over-dispersion in surveillance data by a negative-binomial model. However, in these situations methods like Markov Chain Monte Carlo have to be used in order to obtain the required alarm thresholds, see e.g., [Held et al. \(2006\)](#). A different idea is to capitalise more on SPC techniques, e.g., by extending the CUSUM chart to only specify the parametric form of the alternative and base detection on the generalised-likelihood-ratio statistic ([Höhle 2006](#)). Yet another extension is to provide more complex mechanisms for the simulation of epidemics. In particular it would be interesting to include multi-day outbreaks originating from single-source exposure with varying incubation time ([Hutwagner et al. 2005](#)) or SEIR-like epidemics ([Andersson and Britton 2000](#)).

Acknowledgments My thanks to A. Riebler, C. Lang and M. Paul, University of Munich, for their help on the package implementation. I am grateful to K. Stark and D. Altmann, RKI, C. Dargatz, University of Munich, and A. Zeileis, Wirtschaftsuniversität Wien, for valuable help and discussions. The research was conducted with financial support from the Collaborative Research Centre SFB 386 funded by the German Research Foundation (DFG).

References

- Altmann D (2003) The surveillance system of the Robert Koch Institute, Germany (Personal Communication)
- Andersson H, Britton T (2000) Stochastic epidemic models and their statistical analysis, vol 151. Springer Lectures Notes in Statistics. Springer, Heidelberg
- Ethelberg S, Mølbak K (2007) GastroEnteRitis Monitor, Statens Serum Institut, Denmark, <http://germ.dk> [Online; accessed 27-March-2007]
- Farrington C, Andrews N (2003) Outbreak detection: application to infectious disease surveillance. In: Brookmeyer R, Stroup D (eds), Monitoring the health of populations, chapter 8, pp 203–231. Oxford University Press, NY, USA
- Farrington C, Andrews N, Beale A, Catchpole M (1996) A statistical algorithm for the early detection of outbreaks of infectious disease. *J R Stat Soc Ser A* 159:547–563
- Held L, Hofmann M, Höhle M, Schmid V (2006) A two component model for counts of infectious diseases. *Biostatistics* 7:422–437
- Held L, Höhle M, Hofmann M (2005) A statistical framework for the analysis of multivariate infectious disease surveillance data. *Stat Modell* 5:187–199
- Höhle M (2006) Poisson regression charts for the monitoring of surveillance time series. Technical report, Department of Statistics, University of Munich. SFB Discussion Paper 500
- Hutwagner L, Browne T, Seaman G, Fleischhauer A (2005) Comparing aberration detection methods with simulated data. *Emerg Infect Dis* 11:314–316
- Kenett R, Pollak M (1983) On sequential detection of a shift in the probability of a rare event. *J Am Stat Assoc* 78(382):389–395
- Knoth S (2004) spc: Statistical Process Control. R package version 0.2
- Lawson A, Kleinman K (eds) (2005) Spatial and syndromic surveillance for public health. Wiley, London
- Leisch F (2003) Sweave, Part I: Mixing R and LaTeX. *R Newsletter* 2(3):28–31
- Lewin-Koh NJ, Bivand R, contributions by Edzer Pebesma J, Hausmann P, Rubio VG, Jagger T, Luque SP (2006) maptools: Tools for reading and handling spatial objects. R package version 0.6-5
- Pebesma EJ, Bivand RS (2005) Classes and methods for spatial data in R. *R News* 5(2):9–13
- Pierce D, Schafer D (1986) Residuals in generalized linear models. *J Am Stat Assoc* 81(396):977–986
- R Development Core Team (2006) R: A language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria, ISBN 3-900051-07-0
- Riebler A (2004) Empirischer Vergleich von statistischen Methoden zur Ausbruchserkennung bei Surveillance Daten. Bachelor's thesis, Department of Statistics, University of Munich
- Robert Koch Institute (2004) SurvStat@RKI. <http://www3.rki.de/SurvStat> (Date of query: September 2004)
- Robert Koch Institute (2006) Epidemiologisches Bulletin 33. Available from <http://www.rki.de>
- Rogerson P, Yamada I (2004) Approaches to syndromic surveillance when data consist of small regional counts. *Morb Mortal Wkly Rep* 53:79–85
- Rossi G, Lampugnani L, Marchi M (1999) An approximate CUSUM procedure for surveillance of health events. *Stat Med* 18:2111–2122
- Sonesson C, Bock D (2003) A review and discussion of prospective statistical surveillance in public health. *J R Stat Soc Ser A* 166:5–12
- Sonesson C, Frisén M (2005) Multivariate surveillance. In: Lawson A, Kleinman K (eds) Spatial and syndromic surveillance for public health, chapter 9. Wiley, London, pp 153–166
- Stroup D, Williamson G, Herndon J, Karon J (1989) Detection of aberrations in the occurrence of notifiable diseases surveillance data. *Stat Med* 8:323–329
- Widdowson M-A, Bosman A, van Straten E, Tinga M, Chaves S, van Eerden L, van Pelt W (2003) Automated, Laboratory-based system using the Internet for disease outbreak detection, the Netherlands. *Emerg Infect Dis* 9(9):1046–1052