The function hhh4 in the R-package surveillance

Michaela Paul and Sebastian Meyer*
Division of Biostatistics
Institute of Social and Preventive Medicine
University of Zurich, Switzerland
January 5, 2015

Abstract
This document gives an introduction to the use of the function hhh4 for modelling univariate and multivariate time series of infectious disease counts. The function is part of the R-package surveillance, which provides tools for the visualization, modelling and monitoring of surveillance time series. The basic functionality of surveillance is introduced in the package vignette (Höhle et al., 2007) and in Höhle (2007) with main focus on outbreak detection methods. The following illustrates the use of hhh4 as estimation and prediction routine for the modelling framework proposed by Held et al. (2005), and extended in Paul et al. (2008), Paul and Held (2011) and Herzog et al. (2011).

1 Introduction

To meet the threats of infectious diseases, many countries have established surveillance systems for the reporting of various infectious diseases. The systematic and standardized reporting at a national and regional level aims to recognize all outbreaks quickly, even when aberrant cases are dispersed in space. Traditionally, notification data, i.e. counts of cases confirmed according to a specific definition and reported daily, weekly or monthly on a regional or national level, are used for surveillance purposes. The R-package surveillance provides functionality for the retrospective modelling and prospective change-point detection in the resulting surveillance time series. A recent introduction to the package with focus on outbreak detection methods is given by Höhle and Mazick (2010).

This document illustrates the functionality of the function hhh4 for the modelling of univariate and multivariate time series of infectious disease counts. It is part of the surveillance package as of version 1.3. Section 2 introduces the S4 class data structure used to store surveillance time series data within

*Author of correspondence: Sebastian.Meyer@ifspm.uzh.ch
the package. Access and visualization methods are outlined by means of built-in data sets. In Section 3, the statistical modelling approach by Held et al. (2005) and further model extensions are described. After the general function call and arguments are shown, the detailed usage of hhh4 is demonstrated in Section 4 using data introduced in Section 2.

2 Surveillance data

Denote by \( \{y_{it}; i = 1, \ldots, I, t = 1, \ldots, T\} \) the multivariate time series of disease counts for a specific partition of gender, age and location. Here, \( T \) denotes the length of the time series and \( I \) denotes the number of units (e.g., geographical regions or age groups) being monitored. Such data are represented using objects of the S4 class \texttt{sts} (surveillance time series).

This class contains the \( T \times I \) matrix of counts \( y_{it} \) in a slot \texttt{observed}. An integer slot \texttt{epoch} denotes the time index \( 1 \leq t \leq T \) of each row in \texttt{observed}. The number of observations per year, e.g., 52 for weekly or 12 for monthly data, is denoted by \texttt{freq}. Furthermore, \texttt{start} denotes a vector of length two containing the start of the time series as \( (\text{year}, \text{epoch}) \).

For spatially stratified time series, the slot \texttt{neighbourhood} denotes an \( I \times I \) adjacency matrix with elements 1 if two regions are neighbors and 0 otherwise. For map visualizations, the slot \texttt{map} links the multivariate time series to geographical regions of an ESRI shapefile (using functionality from the package \texttt{maptools} [Lewin-Koh et al., 2010]). Additionally, the slot \texttt{populationFrac} contains a \( T \times I \) matrix representing population fractions in unit \( i \) at time \( t \).

The package \texttt{surveillance} contains a number of time series in the \texttt{data} directory. Most data sets originate from the SurvStat@RKI database[1] maintained by the Robert Koch Institute (RKI), Germany. Selected data sets will be analyzed in Section 4 and are introduced in the following. Note that many of the built-in datasets are stored in the S3 class data structure \texttt{disProg}. They can be easily converted into the S4 \texttt{sts} data structure using the function \texttt{disProg2sts}. The resulting \texttt{sts} object can be accessed similar as standard \texttt{matrix} objects and allows easy temporal and spatial aggregation as will be shown in the remainder of this section.

Example: Influenza and meningococcal disease in Germany 01/2001–52/2006

As a first example, the weekly number of influenza and meningococcal disease cases in Germany is considered.

```r
> # load data
data("influenza")
> # convert to sts class and print basic information about the time series
print(fluMen <- disProg2sts(influenza))
```

-- An object of class sts --

freq: 52
start: 2001 1
dim(observable): 312 2

Head of observed:
influenza meningococcus

head of neighbourhood:
influenza meningococcus

The univariate time series of meningococcal disease counts can be obtained with

> meningo <- fluMen[, "meningococcus"]
> dim(meningo)

[1] 312 1

The plot function provides an interface to the visual representation of the multivariate time series in time, space and space-time which is controlled by the type argument:

> plot(fluMen, type = observed ~ time | unit, # type of plot
+ same.scale = FALSE, # unit-specific ylim ?
+ col = "grey" # color of bars
+ )


Example: Influenza in Southern Germany, 01/2001-52/2008
The spatio-temporal spread of influenza in the 140 Kreise (districts) of Bavaria and Baden-Württemberg is analyzed using the weekly number of cases reported to the RKI (Robert Koch-Institut, 2009) in the years 2001–2008. An sts object containing the data is created as follows:
> # read in observed number of cases
> flu.counts <- as.matrix(read.table(system.file("extdata/counts_flu_BYBW.txt", +     package = "surveillance")))
> # remove 'X' in column names
> colnames(flu.counts) <- substring(colnames(flu.counts), first = 2, last = 5)
> # read in adjacency matrix with elements 1 if two regions share a common border
> nhood <- as.matrix(read.table(system.file("extdata/neighbourhood_BYBW.txt", +     package = "surveillance")))

> # visualize adjacency matrix
> library("Matrix")
> print(image(Matrix(nhood)))

> # read in a shapefile of the districts in Bavaria and Baden-Wuerttemberg
> map <- maptools::readShapePoly(
+     system.file("shapes/districts_BYBW.shp", package = "surveillance"),
+     IDvar = "id"
+ )
> # read in population fractions
> p <- matrix(
+     read.table(system.file("extdata/population_2001-12-31_BYBW.txt", +         package = "surveillance"),
+     header = TRUE)$popFrac,
+     nrow = nrow(flu.counts), ncol = ncol(flu.counts), byrow = TRUE)
> # create sts object
> flu <- new("sts", epoch = 1:nrow(flu.counts),
+     observed = flu.counts,
+     start = c(2001, 1),
+     freq = 52,
+     neighbourhood = nhood,
+     map = map,
+     population = p
+ )

This \texttt{sts} object is already included in \texttt{surveillance} and may be loaded with \texttt{data(fluBYBW)}. 
A map of the total number of cases in the year 2001 may be obtained as follows:

```r
> par(mar=c(0,0,0,0))
> plot(flu[year(flu) == 2001, ],
+     type = observed ~ 1 | unit,
+     labels = FALSE
+ )
```

Example: Measles in Germany, 01/2005–52/2007
The following data set contains the weekly number of measles cases in the 16 German Bundesländer (federal states), in the years 2005–2007. These data are analyzed in [Herzog et al. 2011](herzog2011) after aggregation into successive bi-weekly periods.

```r
> data("measlesDE")
> # aggregate into successive bi-weekly periods
> measles2w <- aggregate(measlesDE, nfreq = 26)
> plot(measles2w, type = observed ~ time,
+     main = "Bi-weekly number of measles cases in Germany",
+     legend.opts = NULL
+ )
```
3 Model formulation

Retrospective surveillance aims to identify outbreaks and (spatio-)temporal patterns through statistical modelling. Motivated by a branching process with immigration, Held et al. (2005) suggest the following model for the analysis of univariate time series of infectious disease counts \( \{y_t; t = 1, \ldots, T\} \). The counts are assumed to be Poisson distributed with conditional mean

\[
\mu_t = \lambda y_{t-1} + \nu_t, \quad (\lambda, \nu_t > 0)
\]

where \( \lambda \) and \( \nu_t \) are unknown quantities. The mean incidence is decomposed additively into two components: an epidemic or autoregressive component \( \lambda y_{t-1} \), and an endemic component \( \nu_t \). The former should be able to capture occasional outbreaks whereas the latter explains a baseline rate of cases with stable temporal pattern. Held et al. (2005) suggest the following parametric model for the endemic component:

\[
\log(\nu_t) = \alpha + \beta t + \left\{ \sum_{s=1}^{S} \gamma_s \sin(\omega_s t) + \delta_s \cos(\omega_s t) \right\}, \quad (1)
\]

where \( \alpha \) is an intercept, \( \beta \) is a trend parameter, and the terms in curly brackets are used to model seasonal variation. Here, \( \gamma_s \) and \( \delta_s \) are unknown parameters, \( S \) denotes the number of harmonics to include, and \( \omega_s = 2\pi s / \text{freq} \) are Fourier frequencies (e.g. \( \text{freq} = 52 \) for weekly data). For ease of interpretation, the seasonal terms in (1) can be written equivalently as

\[
\gamma_s \sin(\omega_s t) + \delta_s \cos(\omega_s t) = A_s \sin(\omega_s t + \varphi_s)
\]

with amplitude \( A_s = \sqrt{\gamma_s^2 + \delta_s^2} \) describing the magnitude, and phase difference \( \tan(\varphi_s) = \delta_s / \gamma_s \) describing the onset of the sine wave.

To account for overdispersion, the Poisson model may be replaced by a negative binomial model. Then, the conditional mean \( \mu_t \) remains the same but
the conditional variance increases to \( \mu_t(1 + \mu_t\psi) \) with additional unknown overdispersion parameter \( \psi > 0 \).

The model is extended to multivariate time series \( \{y_{it}\} \) in Held et al. (2005) and Paul et al. (2008) by including an additional neighbor-driven component, where past cases in other (neighboring) units also enter as explanatory covariates. The conditional mean \( \mu_{it} \) is then given by

\[
\mu_{it} = \lambda y_{i,t-1} + \phi \sum_{j \neq i} w_{ji} y_{j,t-1} + e_{it}\nu_t, \tag{2}
\]

where the unknown parameter \( \phi \) quantifies the influence of other units \( j \) on unit \( i \), \( w_{ji} \) are suitably chosen known weights and \( e_{it} \) corresponds to an offset (such as population fractions at time \( t \) in region \( i \)). A simple choice for the weights is \( w_{ji} = 1 \) if units \( j \) and \( i \) are adjacent and 0 otherwise. See Paul et al. (2008) for a discussion of alternative weights.

When analyzing a specific disease observed in, say, multiple regions or several pathogens (such as influenza and meningococcal disease), the assumption of equal incidence levels or disease transmission across units is questionable. To address such heterogeneity, the unknown quantities \( \lambda, \phi, \) and \( \nu_t \) in (2) may also depend on unit \( i \). This can be done via

- unit-specific fixed parameters, e.g. \( \log(\lambda_i) = \alpha_i \) (Paul et al., 2008);
- unit-specific random effects, e.g. \( \log(\lambda_i) = \alpha_0 + a_i, \ a_i \sim N(0, \sigma^2) \) (Paul and Held, 2011);
- linking parameters with known (possibly time-varying) explanatory variables, e.g. \( \log(\lambda_i) = \alpha_0 + x_i\alpha_1 \) with region-specific vaccination coverage \( x_i \) (Herzog et al., 2011).

A call to \texttt{hhh4} fits a Poisson or negative binomial model with conditional mean

\[
\mu_{it} = \lambda_{it} y_{i,t-1} + \phi_{it} \sum_{j \neq i} w_{ji} y_{j,t-1} + e_{it}\nu_{it}
\]

to a multivariate time series of counts. Here, the three unknown quantities are decomposed additively on the log scale

\[
\begin{align*}
\log(\lambda_{it}) &= \alpha_0 + a_i + u_{it}^T \alpha \quad \text{(ar)} \\
\log(\phi_{it}) &= \beta_0 + b_i + x_{it}^T \beta \quad \text{(ne)} \\
\log(\nu_{it}) &= \gamma_0 + c_i + z_{it}^T \gamma \quad \text{(end)}
\end{align*}
\]

where \( \alpha_0, \beta_0, \gamma_0 \) are intercepts, \( \alpha, \beta, \gamma \) are vectors of unknown parameters corresponding to covariate vectors \( u_{it}, x_{it}, z_{it} \), and \( a_i, b_i, c_i \) are random effects. For instance, model (1) with \( S = 1 \) seasonal terms may be represented
as \( z_{it} = (t, \sin(2\pi/\text{freq } t), \cos(2\pi/\text{freq } t))^\top \). The stacked vector of all random effects is assumed to follow a normal distribution with mean 0 and covariance matrix \( \Sigma \), see [Paul and Held (2011)] for further details. Inference is based on (penalized) likelihood methodology as proposed in [Paul and Held (2011)]. In applications, each component (ar)-(end) may be omitted in parts or as a whole.

4 Function call and control settings

The estimation procedure is called with

\>
\texttt{hhh4(sts, control)}
\>

where \texttt{sts} denotes a (multivariate) surveillance time series and the model is specified in the argument \texttt{control} in consistency with other algorithms in \texttt{surveillance}. The \texttt{control} setting is a list of the following arguments:

\>
\texttt{control = list(}
+ \texttt{ar = list(f = “-1”),} # formula: \( \exp(u'alpha) \ast y_{i,t-1} \)
+ \texttt{ne = list(f = “-1”),} # formula: \( \exp(x'beta) \ast \sum_j \{w_{ji} \ast y_{j,t-1}\} \)
+ \texttt{weights = neighbourhood(stsObj)),} # matrix of weights \( w_{ji} \)
+ \texttt{end = list(f = “1”),} # formula: \( \exp(z'gamma) \ast e_{it} \)
+ \texttt{offset = 1),} # optional offset \( e_{it} \)
+ \texttt{family = “Poisson”,} # Poisson or NegBin model
+ \texttt{subset = 2:nrow(stsObj),} # subset of observations to be used
+ \texttt{optimizer = list(),} # control optimization procedure
+ \texttt{verbose = FALSE,} # no progress information is printed
+ \texttt{start = list(fixed = NULL,} # list with initial values for fixed,
+ \texttt{random = NULL,} # random, and
+ \texttt{sd.corr = NULL),} # variance parameters
+ \texttt{data = list(t=epoch(stsObj)-1),} # named list of covariates
+ \texttt{keep.terms = FALSE} # do not keep the model terms
+ \texttt{)}
\>

The first three arguments \texttt{ar}, \texttt{ne}, and \texttt{end} specify the model components using formula objects. As default, the counts \( y_{it} \) are assumed to be Poisson distributed. A negative binomial model is obtained with \texttt{family = “NegBin1”}. By default, both the penalized and marginal log-likelihoods are maximized using the optimization routine implemented in \texttt{nlminb}. The methods implemented in \texttt{optim} may also be used, e.g. \texttt{optimizer = list(variance = list(method=”Nelder-Mead")} is an attractive alternative for maximization of the marginal log-likelihood with respect to the variance parameters (see \texttt{?hhh4}). Initial values for the fixed, random, and variance parameters are passed on in the \texttt{start} argument. If the model contains covariates, these have to be specified in the \texttt{data} argument. When covariates do not vary across units, they may be passed on as a vector of length \( T \). Otherwise, covariate values have to be stored and passed on in a matrix of size \( T \times I \).
In the following, the functionality of \texttt{hhh4} is demonstrated using the data sets introduced in Section 2 and previously analyzed in Paul et al. (2008), Paul and Held (2011) and Herzog et al. (2011). Selected results are reproduced. For a thorough discussion we refer to these papers.

Univariate modelling

As a first example, consider the univariate time series of meningococcal infections in Germany, 01/2001–52/2006 (cf. Tab. 1 in Paul et al., 2008). A Poisson model without autoregression and $S = 1$ seasonal term is specified as follows:

\begin{verbatim}
> # specify formula object for endemic component
> (f_S1 <- addSeason2formula(f = ~ 1, S = 1, period = 52))

\end{verbatim}

\begin{verbatim}
~1 + sin(2 * pi * t/52) + cos(2 * pi * t/52)
\end{verbatim}

\begin{verbatim}
> # fit Poisson model
> summary(hhh4(meningo, control = list(end = list(f = f_S1), family = "Poisson")))
\end{verbatim}

Call:
\texttt{hhh4(stsObj = meningo, control = list(end = list(f = f_S1), family = "Poisson"))}

Coefficients:
\begin{tabular}{lrr}
end.1 & 2.26478 & 0.01871 \\
end.sin(2 * pi * t/52) & 0.36195 & 0.02590 \\
end.cos(2 * pi * t/52) & 0.26055 & 0.02578 \\
\end{tabular}
Log-likelihood: -872.09
AIC: 1750.19
BIC: 1761.41

Number of units: 1
Number of time points: 311

A corresponding negative binomial model is obtained via

\begin{verbatim}
> result1 <- hhh4(meningo, control = list(end = list(f = f_S1),
+ family = "NegBin1"))
\end{verbatim}

As default, the autoregressive component is omitted with $\sim -1$ in the formula specification. In can be included in the model with

\begin{verbatim}
> m2 <- list(ar = list(f = ~ 1), # log(lambda) = alpha
+ end = list(f = f_S1),
+ family = "NegBin1",
+ # use estimates from previous model as initial values
+ start = list(fixed = c(log(0.1), # initial values for alpha,
+ coef(result1)) # and remaining parameters
+ )
+ )
\end{verbatim}
> # fit model
> result2 <- hhh4(meningo, control = m2)
> # extract ML estimates
> round(coef(result2, se = TRUE, idx2Exp = 1), 2)

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>exp((ar.1))</td>
<td>0.16</td>
<td>0.06</td>
</tr>
<tr>
<td>end.1</td>
<td>2.09</td>
<td>0.07</td>
</tr>
<tr>
<td>end.sin((2 * pi * t/52))</td>
<td>0.34</td>
<td>0.04</td>
</tr>
<tr>
<td>end.cos((2 * pi * t/52))</td>
<td>0.26</td>
<td>0.04</td>
</tr>
<tr>
<td>overdisp</td>
<td>0.05</td>
<td>0.01</td>
</tr>
</tbody>
</table>

> # get AIC
> AIC(result2)

[1] 1701.228

Bivariate modelling

Now, the weekly numbers of both meningococcal disease (MEN) and influenza (FLU) cases are analyzed to investigate whether influenza infections predispose meningococcal disease (cf. Tab. 2 in Paul et al., 2008). This requires disease-specific parameters which are specified in the formula object with \(fe(\ldots)\). In the following, a negative binomial model with mean

\[
\begin{pmatrix}
\mu_{\text{men},t} \\
\mu_{\text{flu},t}
\end{pmatrix} = \begin{pmatrix}
\lambda_{\text{men}} & \phi \\
0 & \lambda_{\text{flu}}
\end{pmatrix}
\begin{pmatrix}
\text{MEN}_{t-1} \\
\text{FLU}_{t-1}
\end{pmatrix} + \begin{pmatrix}
\nu_{\text{men},t} \\
\nu_{\text{flu},t}
\end{pmatrix},
\]

where the endemic component includes \(S = 3\) seasonal terms for the FLU data and \(S = 1\) seasonal terms for the MEN data is considered. Here, \(\phi\) quantifies the influence of past influenza cases on the meningococcal disease incidence. This model corresponds to the second model of Tab. 2 in Paul et al. (2008) and is fitted as follows:

> # no "transmission" from meningococcus to influenza
> neighbourhood(fluMen) [\"meningococcus\",\"influenza\"] <- 0
> neighbourhood(fluMen)

<table>
<thead>
<tr>
<th></th>
<th>meningococcus</th>
<th>influenza</th>
</tr>
</thead>
<tbody>
<tr>
<td>meningococcus</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>influenza</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
> # create formula for endemic component
> f.end <- addSeason2formula(f = `~-1 + fe(1, which = c(TRUE, TRUE)),
+   # disease-specific intercepts
+   S = c(3, 1),  # S = 3 for flu, S = 1 for men
+   period = 52)
> # specify model
> m <- list(ar = list(f = `~-1 + fe(1, which=c(TRUE, TRUE))),
+   ne = list(f = `~-1 + fe(1, which=c(FALSE, TRUE))),
+   end = list(f = f.end),
+   family = "NegBinM"
+ )
> # fit model
> summary(result <- hhh4(fluMen, control = m))

Call:
hhh4(stsObj = fluMen, control = m)

Coefficients:

            Estimate Std. Error
ar.1.influenza  -0.30436  0.06783
ar.1.meningococcus  -2.35234  0.59796
ne.1.meningococcus  -5.21667  0.26049
end.1.influenza     1.08829  0.16532
end.1.meningococcus  2.11860  0.06683
end.sin(2 * pi * t/52).influenza  1.18619  0.23598
end.sin(2 * pi * t/52).meningococcus  0.26661  0.03968
end.cos(2 * pi * t/52).influenza     1.50978  0.14671
end.cos(2 * pi * t/52).meningococcus  0.22904  0.03532
end.sin(4 * pi * t/52).influenza      0.91917  0.17150
end.cos(4 * pi * t/52).influenza     -0.16160  0.17985
end.sin(6 * pi * t/52).influenza     0.36924  0.14998
end.cos(6 * pi * t/52).influenza    -0.53455  0.16190
overdisp.influenza    0.29455  0.03577
overdisp.meningococcus  0.03950  0.01089

Log-likelihood:  -1880.97
AIC:            3791.94
BIC:           3858.43

Number of units:  2
Number of time points:  311

A plot of the estimated mean for the meningococcal disease data, decomposed into the three components, is obtained with

> plot(result, units = "meningococcus")
Multivariate modelling

For disease counts observed in a large number of regions, say, (i.e. highly multivariate time series of counts) the use of region-specific parameters to account for regional heterogeneity is no longer feasible, as estimation and identifiability problems may occur. Paul and Held (2011) propose a random effects formulation to analyze the weekly number of influenza cases in 140 districts of Southern Germany. For example, consider a model with random intercepts in the endemic component: \( c_i \sim N(0, \sigma^2_i) \), \( i = 1, \ldots, I \). Such effects are specified in a formula object as

\[
> f.end <- ~ -1 + ri(type = "iid", corr = "all")
\]

Setting type = "car" would assume that the random effects are spatially correlated instead of uncorrelated. See Paul and Held (2011) for further details. The argument corr = "all" allows for correlation between region-specific random effects in different components, e.g. random incidence levels \( c_i \) in the endemic component and random effects \( b_i \) in the neighbor-driven component. The following call to \texttt{hhh4} fits such a random effects model with linear trend and \( S = 3 \) seasonal terms in the endemic component and a fixed autoregressive parameter \( \lambda \) to the influenza data (cf. model B2 in Tab. 3 in Paul and Held [2011]).

\[
> # weight matrix \ w_{ji} = 1/(No. neighbors of j) if j \sim i, and 0 otherwise
> wji <- neighbourhood(flu)/rowSums(neighbourhood(flu))
> # endemic component: iid random effects, linear trend, and \( S=3 \) seasonal terms
> f.end <- addSeason2formula(f = ~ -1 + ri(type = "iid", corr="all") +
+ I((t-208)/100),
+ S = 3,
+ period = 52)
> model.B2 <- list(ar = list(f = ~ 1),
+ ne = list(f = ~ -1+ ri(type = "iid", corr="all"),
+ weights = wji))
\]
Call:

```
```

Random effects:

```
Var Corr
ne.ri(iid) 0.9594
end.ri(iid) 0.5094 0.5617
```

Fixed effects:

```
Estimates Std. Error
ar.1 -0.89756 0.03688
ne.ri(iid) -1.52555 0.10353
end.1((t - 208)/100) 0.56202 0.02354
end.sin(2 * pi * t/52) 2.18487 0.09854
end.cos(2 * pi * t/52) 2.33195 0.12238
end.sin(4 * pi * t/52) 0.44030 0.10525
end.cos(4 * pi * t/52) -0.39470 0.09402
end.sin(6 * pi * t/52) 0.32173 0.06479
end.cos(6 * pi * t/52) -0.26468 0.06308
end.ri(iid) 0.21924 0.10281
overdisp 1.09909 0.03429
```

Penalized log-likelihood:  -18742.42
Marginal log-likelihood:  -343.26

Number of units:  140
Number of time points:  416

Model choice based on information criteria such as AIC or BIC is well explored and understood for models that correspond to fixed-effects likelihoods. However, in the presence of random effects their use can be problematic. For model selection in time series models, the comparison of successive one-step-ahead forecasts with the actually observed data provides a natural alternative. In this context, Gneiting and Raftery [2007] recommend the use of strictly proper scoring rules, such as the logarithmic score or the ranked probability score. See Czado et al. [2009] and Paul and Held [2011] for further details.

One-step-ahead predictions for the last 2 years for model B2 are obtained as follows:

```
```

The mean logarithmic and mean ranked probability score are then computed with

```
> colMeans(scores(pred.B2)[, c("logs", "rps")])
```

```
  logs  rps
0.5632647 0.4362529
```
As a last example, consider the number of measles cases in the 16 federal states of Germany, in the years 2005–2007. There is considerable regional variation in the incidence pattern which is most likely due to differences in vaccination coverage. In the following, information about vaccination coverage in each state, namely the log proportion of unvaccinated school starters, is included as explanatory variable in a model for the bi-weekly aggregated measles data. See [Herzog et al. (2011)] for further details. Vaccination coverage levels for the year 2006 are available in the dataset data(MMRcoverageDE). This dataset can be used to compute the $78 \times 16$ matrix vac0 with adjusted proportions of unvaccinated school starters in each state $i$ used by [Herzog et al. (2011)]

```r
> vac0[1:2, 1:5]

<table>
<thead>
<tr>
<th></th>
<th>Baden-Wuerttemberg</th>
<th>Bavaria</th>
<th>Berlin</th>
<th>Brandenburg</th>
<th>Bremen</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, )</td>
<td>0.1000115</td>
<td>0.113261</td>
<td>0.099989</td>
<td>0.0605575</td>
<td>0.115963</td>
</tr>
<tr>
<td>(2, )</td>
<td>0.1000115</td>
<td>0.113261</td>
<td>0.099989</td>
<td>0.0605575</td>
<td>0.115963</td>
</tr>
</tbody>
</table>
```

A Poisson model which links the autoregressive parameter with this covariate and contains $S = 1$ seasonal term in the endemic component (cf. model A0 in Tab. 3 in [Herzog et al. 2011]) is obtained with

```r
> # endemic component: Intercept + $S = 1$ sine/cosine pair
> f.end <- addSeason2formula(f = ~ 1, S = 1, period = 26)
> # autoregressive component: Intercept + vaccination coverage information
> model.A0 <- list(ar = list(f = ~ 1 + logVac0),
>   end = list(f = f.end, offset = population(measles2w)),
>   data = list(t = epoch(measles2w), logVac0 = log(vac0)))
> # fit model
> result.A0 <- hhh4(measles2w, model.A0)
> # parameter estimates
> round(coef(result.A0, se = TRUE, amplitudeShift = TRUE), 2)
```

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar.1</td>
<td>3.01</td>
</tr>
<tr>
<td>ar.logVac0</td>
<td>1.38</td>
</tr>
<tr>
<td>end.1</td>
<td>1.78</td>
</tr>
<tr>
<td>end.A(2 * pi * t/26)</td>
<td>0.66</td>
</tr>
<tr>
<td>end.s(2 * pi * t/26)</td>
<td>-0.10</td>
</tr>
</tbody>
</table>

5 Summary

As part of the R-package surveillance, the function hhh4 provides a flexible tool for the modelling of multivariate time series of infectious disease counts. The discussed count data model is able to account for serial and spatio-temporal correlation, as well as heterogeneity in incidence levels and disease transmission. The functionality of hhh4 was illustrated using several built-in data sets.
References


