The function 'hhh4' in the R-package 'surveillance'

Michaela Paul*
Biostatistics Unit
Institute of Social and Preventive Medicine
University of Zurich, Switzerland

February 1, 2011

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 (2010) and Herzog et al. (2010).

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. The function is currently incorporated in the development version of `surveillance` available from `http://surveillance.r-forge.r-project.`

*Author of correspondence: Email: `michaela.paul@ifspm.uzh.ch`

org/. Section 2 introduces the S4 class data structure used to store surveillance time series data within 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 `sts` (surveillance time series).
This class contains the $T \times I$ matrix of counts y_{it} in a slot `observed`. An integer slot `epoch` denotes the time index $1 \leq t \leq T$ of each row in `observed`. The number of observations per year, e.g. 52 for weekly or 12 for monthly data, is denoted by `freq`. Furthermore, `start` denotes a vector of length two containing the start of the time series as `c(year, epoch)`. For spatially stratified time series, the slot `neighbourhood` denotes an $I \times I$ adjacency matrix with elements 1 if two regions are neighbors and 0 otherwise. For map visualizations, the slot `map` links the multivariate time series to geographical regions of an ESRI shapefile (using functionality from the package `maptools` (Lewin-Koh et al., 2010)). Additionally, the slot `populationFrac` contains a $T \times I$ matrix representing population fractions in unit i at time t.
The package `surveillance` contains a number of time series in the `data` directory. Most data sets originate from the SurvStat@RKI database (`http://www3.rki.de/SurvStat`), 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 `disProg`. They can be easily converted into the S4 `sts` data structure using the function `disProg2sts`. The resulting `sts` object can be accessed similar as standard `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.

```
> # load data
> data("influMen")
```

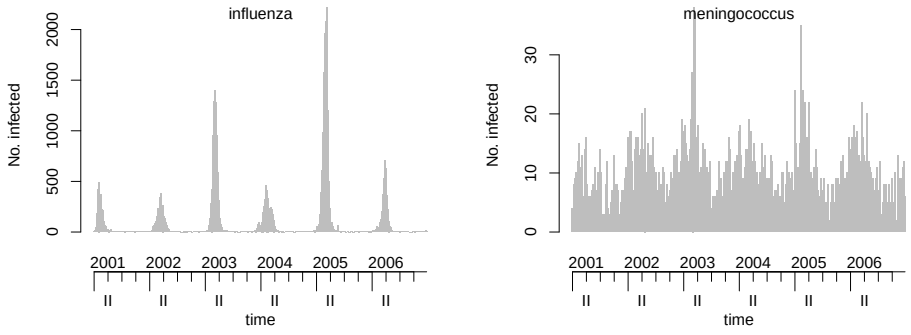
```
> # convert to sts class and print basic information about the time series
> print(fluMen <- disProg2sts(influMen))


-- An object of class sts --
freq:                  52
start:                  2001 1
dim(observed):          312 2

Head of observed:
     influenza meningococcus
[1,]         7             4

map:
NULL

head of neighbourhood:
           influenza meningococcus
influenza          0             1
```

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:

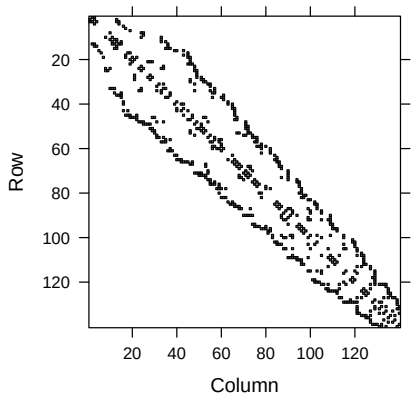
```
> plot(fluMen, type = observed ~ time | unit, # type of plot
+              same.scale = FALSE,            # unit-specific ylim ?
+              col = "grey"                   # color of bars
+              )
```



See Höhle and Mazick (2010) for a detailed description of the plot routines.

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("data/flu_ByBw.txt",package="surveillance")))
> # read in adjacency matrix with elements 1 if two regions share a common border
> nhood <- as.matrix(read.table(system.file("data/neighbourhood_ByBw.txt",package="surveillance")))
> # visualize adjacency matrix
> image(Matrix(nhood))
```

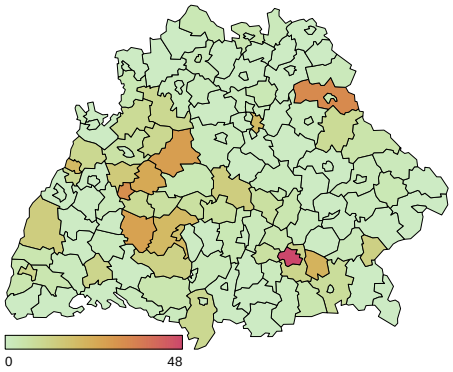


Dimensions: 140 x 140

```
> # read in a shapefile of the districts in Bavaria and Baden-Wuerttemberg
> map <- readShapePoly(system.file("shapes/districts_BYBW.shp",package="surveillance"), IDvar = "id")
> # read in population fractions
> p <- as.matrix(read.table(system.file("data/population_2001-12-31_ByBw.txt",package="surveillance"))
> # 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
+                   )
```

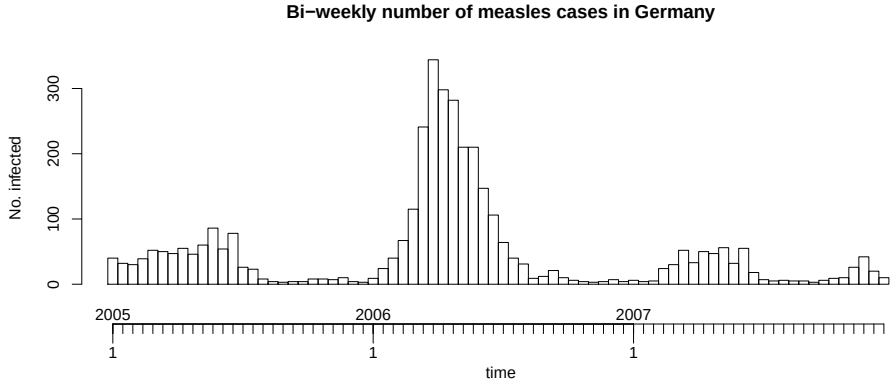
A map of the total number of cases in the year 2001 may be obtained as follows:

```
> plot(flu[year(flu) == 2001, ],    # select year 2001
+      type = observed ~ 1 | unit,  # map of counts aggregated over times t
+      labels = FALSE               # suppress region labels in map
+      )
```



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. (2010) after aggregation into successive bi-weekly periods.

```
> data("measlesDE")
> # aggregate into successive bi-weekly periods
> measles2w <- aggregate(measles.land, nfreq = 26)
> plot(measles2w, type = observed ~ time,   # ts plot aggregated over all units i
+                 main = "Bi-weekly number of measles cases in Germany",
+                 legend.opts = NULL        # suppress default legend
+                 )
```



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 λ and ν_t are unknown quantities. The mean incidence is decomposed additively into two components: an epidemic or *autoregressive* component λy_{t-1}, and an *endemic* component ν_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:

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

where α is an intercept, β is a trend parameter, and the terms in curly brackets are used to model seasonal variation. Here, γ_s and δ_s are unknown parameters, S denotes the number of harmonics to include, and $\omega_s = 2\pi s/\texttt{freq}$ are Fourier frequencies (e.g. `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 μ_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 μ_{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 ϕ 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 λ, ϕ, and ν_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 $\log(\lambda_i) = \alpha_0 + a_i$, $a_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2_\lambda)$ (Paul and Held, 2010);
- 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., 2010).

A call to `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

$$\log(\lambda_{it}) = \alpha_0 + a_i + \boldsymbol{u}_{it}^\top \boldsymbol{\alpha} \qquad (\texttt{ar})$$
$$\log(\phi_{it}) = \beta_0 + b_i + \boldsymbol{x}_{it}^\top \boldsymbol{\beta} \qquad (\texttt{ne})$$
$$\log(\nu_{it}) = \gamma_0 + c_i + \boldsymbol{z}_{it}^\top \boldsymbol{\gamma} \qquad (\texttt{end})$$

where $\alpha_0, \beta_0, \gamma_0$ are intercepts, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are vectors of unknown parameters corresponding to covariate vectors $\boldsymbol{u}_{it}, \boldsymbol{x}_{it}, \boldsymbol{z}_{it}$, and a_i, b_i, c_i are random effects. For instance, model (1) with $S = 1$ seasonal terms may be represented as $\boldsymbol{z}_{it} = (t, \sin(2\pi/\texttt{freq}\, t), \cos(2\pi/\texttt{freq}\, t))^\top$. The stacked vector of all random effects is assumed to follow a normal distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$, see Paul and Held (2010) for further details. Inference is based on (penalized) likelihood methodology as proposed in Paul and Held (2010). 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

```
> hhh4(sts, control)
```

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

```
> control = list(
+     ar = list(f = ~ -1),        # formula: exp(u'alpha) * y_i,t-1
+     ne = list(f = ~ -1,         # formula: exp(x'beta) * sum_j {w_ji * y_j,t-1}
+               weights = NULL    # matrix with weights w_ji
+                                 # [w_ji = neighbourhood(stsObj) as default]
```

```
+              ),
+     end = list(f = ~ 1,         # formula:  exp(z'gamma) * e_it
+                offset = NULL    # optional offset e_it
+                ),
+     family = "Poisson",                # Poisson or NegBin model
+     subset = 2:nrow(stsObj),           # subset of observations to be used
+                                        # in the fitting process
+     optimizer = list(tech = "nlminb"), # details for optimizer
+     verbose = FALSE,                   # no progress information is printed
+     start = list(fixed = NULL,         # list with initial values for fixed,
+                  random = NULL,        # random, and
+                  sd.corr = NULL        # variance parameters
+                  ),
+     data = data.frame(t = epoch(sts))  # data.frame,
+                                        # or named list with covariates
+     )
```

The first three arguments `ar`, `ne`, and `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 `family = "NegBin1"`. The log-likelihood is maximized using the optimization routine implemented in `nlminb`. Alternatively, the methods implemented in `optim` may be used, e.g. `optimizer = list(tech = "BFGS")`. Initial values for the fixed, random, and variance parameters are passed on in the `start` argument. If the model contains covariates, these have to be specified in the `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 `hhh4` is demonstrated using the data sets introduced in Section 2 and previously analyzed in Paul et al. (2008), Paul and Held (2010) and Herzog et al. (2010). 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:

```
> # specify formula object for endemic component
> ( f_S1 <- addSeason2formula(f = ~ 1, S = 1, period = 52) )

~1 + sin(2 * pi * t/52) + cos(2 * pi * t/52)

> # fit Poisson model
> hhh4(meningo, control = list(end = list(f = f_S1), family = "Poisson"))
```

```
Call:
hhh4(stsObj = meningo, control = list(end = list(f = f_S1), family = "Poisson"))


Fixed effects:
                        Estimates  Std.Error
end.1                      2.2648     0.0187
end.sin(2 * pi * t/52)     0.3619     0.0259
end.cos(2 * pi * t/52)     0.2605     0.0258

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

```
> result1 <- hhh4(meningo, control = list(end = list(f = f_S1),
+                                         family = "NegBin1"))
```

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

```
> 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
+                         )
+            )
> # fit model
> result2 <- hhh4(meningo, control = m2)
> # extract ML estimates
> round(coef(result2, se = TRUE,      # also return standard errors
+                     idx2Exp = 1     # exponentiate 1st param [-> exp(alpha)]
+                     ),2)


                       Estimates Std. Error
exp(ar.1)                  0.16       0.06
end.1                      2.09       0.07
end.sin(2 * pi * t/52)     0.34       0.04
end.cos(2 * pi * t/52)     0.26       0.04
1/overdisp                 0.05       0.01


> # 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(...)`. In the following, a negative binomial model with mean

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

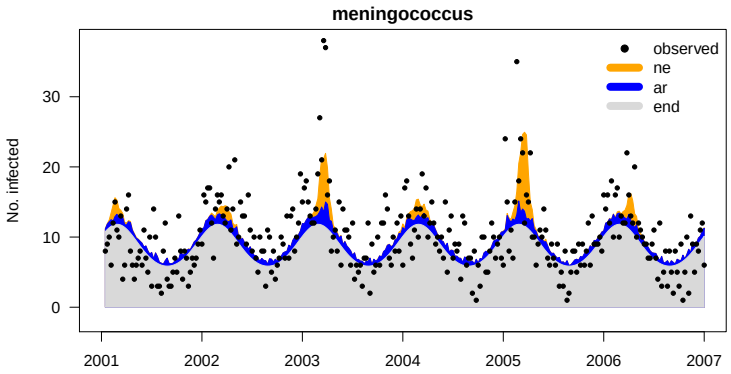
where the endemic component includes $S = 3$ seasonal terms for the FLU data and $S = 1$ seasonal terms for the MEN data is considered. Here, ϕ quantifies the influence of past influenza cases on the meningococcal disease incidence. This model corresponds to the second model of Tab. 2 in Paul et al.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

```
> # 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
> (result <- hhh4(fluMen, control = m))


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


Fixed effects:
                                      Estimates  Std.Error
ar.1.influenza                          -0.3044     0.0678
ar.1.meningococcus                      -2.3523     0.5980
ne.1.meningococcus                      -5.2167     0.2605
end.1.influenza                          1.0883     0.1653
end.1.meningococcus                      2.1186     0.0668
end.sin(2 * pi * t/52).influenza         1.1862     0.2360
end.sin(2 * pi * t/52).meningococcus     0.2666     0.0397
end.cos(2 * pi * t/52).influenza         1.5098     0.1467
end.cos(2 * pi * t/52).meningococcus     0.2290     0.0353
end.sin(4 * pi * t/52).influenza         0.9192     0.1715
end.cos(4 * pi * t/52).influenza        -0.1616     0.1799
end.sin(6 * pi * t/52).influenza         0.3692     0.1500
end.cos(6 * pi * t/52).influenza        -0.5345     0.1619
1/overdisp.influenza                     0.2946     0.0358
1/overdisp.meningococcus                 0.0395     0.0109

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, i = 2, col = c("orange", "blue", "grey85"), legend = TRUE)
```



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 (2010) 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: $c_i \sim \mathcal{N}(0, \sigma_\nu^2), i = 1, \ldots, I$. 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 (2010) 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 `hhh4` fits such a random effects model with linear trend and $S = 3$ seasonal terms in the endemic component and a fixed autoregressive parameter λ to the influenza data (cf. model B2 in Tab. 3 in Paul and Held, 2010).

```
> # weight matrix w_ji = 1/(No. neighbors of j) if j ~ 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),
+                  end = list(f = f.end, offset = population(flu)),
+                  family = "NegBin1"
+                  )
> # fit model
> (result.B2 <- hhh4(flu, model.B2))


Call:
hhh4(stsObj = flu, control = model.B2)

Random effects:
             Var    Corr
ne.ri(iid)  0.9594
end.ri(iid) 0.5094 0.5617


Fixed effects:
                         Estimates  Std.Error
ar.1                       -0.8976     0.0369
ne.ri(iid)                 -1.5256     0.1035
end.I((t - 208)/100)        0.5620     0.0235
end.sin(2 * pi * t/52)      2.1849     0.0985
end.cos(2 * pi * t/52)      2.3319     0.1224
end.sin(4 * pi * t/52)      0.4403     0.1053
end.cos(4 * pi * t/52)     -0.3947     0.0940
end.sin(6 * pi * t/52)      0.3217     0.0648
end.cos(6 * pi * t/52)     -0.2647     0.0631
end.ri(iid)                 0.2192     0.1028
1/overdisp                  1.0991     0.0343

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 (2010) for further details.

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

```
> pred.B2 <- oneStepAhead(result.B2, tp = nrow(flu) - 2 * 52)
```

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. (2010) for further details.
The 78×16 matrix `vac0` contains the proportion of unvaccinated school starters in each state i.

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

```
     Baden-Wrttemberg  Bavaria   Berlin Brandenburg   Bremen
[1,]        0.1000115 0.113261 0.099989   0.0605575 0.115963
[2,]        0.1000115 0.113261 0.099989   0.0605575 0.115963
```

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., 2010) is obtained with

```
> # 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,              # also return standard errors
+            amplitudeShift = TRUE   # transform sin/cos terms to
+            ), 2)                   # Amplitude/shift formulation
```

```
                     Estimates Std. Error
ar.1                      3.01       0.52
ar.logVac0                1.38       0.23
end.1                     1.78       0.06
end.A(2 * pi * t/26)      0.66       0.08
end.s(2 * pi * t/26)     -0.10       0.12
```

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

Czado, C., Gneiting, T., and Held, L. (2009). Predictive model assessment for count data. *Biometrics*, 65(4):1254–1261.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Held, L., Höhle, M., and Hofmann, M. (2005). A statistical framework for the analysis of multivariate infectious disease surveillance counts. *Statistical Modelling*, 5(3):187–199.

Herzog, S. A., Paul, M., and Held, L. (2010). Heterogeneity in vaccination coverage explains the size and occurrence of measles epidemics in German surveillance data. *Epidemiology and Infection*. doi:10.1017/S0950268810001664.

Höhle, M. (2007). Surveillance: an R package for the monitoring of infectious diseases. *Computational Statistics*, 22(4):571–582.

Höhle, M. and Mazick, A. (2010). Aberration detection in R illustrated by Danish mortality monitoring. In Kass-Hout, T. and Zhang, X., editors, *Biosurveillance: A Health Protection Priority*. CRC Press.

Höhle, M., Riebler, A., and Paul, M. (2007). The R-package 'surveillance'.

Lewin-Koh, N. J., Bivand, R., contributions by Edzer J. Pebesma, Archer, E., Baddeley, A., Bibiko, H.-J., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Rubio, V. G., Hausmann, P., Jagger, T., Luque, S. P., MacQueen, D., Niccolai, A., Short, T., and Stabler, B. (2010). *maptools: Tools for reading and handling spatial objects*. R package version 0.7-34.

Paul, M. and Held, L. (2010). Predictive assessment of a non-linear random effects model for multivariate time series of infectious disease counts. Submitted to *Statistics in Medicine*.

Paul, M., Held, L., and Toschke, A. M. (2008). Multivariate modelling of infectious disease surveillance data. *Statistics in Medicine*, 27(29):6250–6267.

Robert Koch-Institut (2009). SurvStat@RKI. `http://www3.rki.de/SurvStat`. Accessed March 2009.