This introduction to the `twinstim` modeling framework of the R package **surveillance** is based on a publication in the *Journal of Statistical Software* – Meyer, Held, and Höhle (2017, Section 3) – which is the suggested reference if you use the `twinstim` implementation in your own work.

`twinstim`: An endemic-epidemic modeling framework for spatio-temporal point patterns

Sebastian Meyer[*]
Friedrich-Alexander-Universität Erlangen-Nürnberg

Leonhard Held
University of Zurich

Michael Höhle
Stockholm University

Abstract

The availability of geocoded health data and the inherent temporal structure of communicable diseases have led to an increased interest in statistical models and software for spatio-temporal data with epidemic features. The R package **surveillance** can handle various levels of aggregation at which infective events have been recorded. This vignette illustrates the analysis of *point-referenced* surveillance data using the endemic-epidemic point process model "`twinstim`" proposed by Meyer, Elias, and Höhle (2012) and extended in Meyer and Held (2014). We first describe the general modeling approach and then exemplify data handling, model fitting, visualization, and simulation methods for time-stamped geo-referenced case reports of invasive meningococcal disease (IMD) caused by the two most common bacterial finetypes of meningococci in Germany, 2002–2008.

Keywords: spatio-temporal point pattern, endemic-epidemic modeling, infectious disease epidemiology, self-exciting point process, spatial interaction function, branching process with immigration.

1. Model class: `twinstim`

Infective events occur at specific points in continuous space and time, which gives rise to a spatio-temporal point pattern $\{(\boldsymbol{s}_i, t_i) : i = 1, \ldots, n\}$ from a region $\boldsymbol{W}$ observed during a period $(0, T]$. The locations $\boldsymbol{s}_i$ and time points t_i of the n events can be regarded as a realization of a self-exciting spatio-temporal point process, which can be characterized by its conditional intensity function (CIF, also termed intensity process) $\lambda(\boldsymbol{s}, t)$. It represents the instantaneous event rate at location $\boldsymbol{s}$ at time point t given all past events, and is often more verbosely denoted by λ^* or by explicit conditioning on the "history" $\mathcal{H}_t$ of the process. Daley and Vere-Jones (2003, Chapter 7) provide a rigorous mathematical definition of this concept, which is key to likelihood analysis and simulation of "evolutionary" point processes.

Meyer *et al.* (2012) formulated the model class "`twinstim`" – a *two*-component *s*patio-*t*emporal *i*ntensity *m*odel – by a superposition of an endemic and an epidemic component:

[*]Author of correspondence: `seb.meyer@fau.de`

$$\lambda(\boldsymbol{s},t) = \nu_{[\boldsymbol{s}][t]} + \sum_{j \in I(\boldsymbol{s},t)} \eta_j \, f(\|\boldsymbol{s} - \boldsymbol{s}_j\|) \, g(t - t_j) \, . \tag{1}$$

This model constitutes a branching process with immigration. Part of the event rate is due to the first, endemic component, which reflects sporadic events caused by unobserved sources of infection. This background rate of new events is modeled by a log-linear predictor $\nu_{[\boldsymbol{s}][t]}$ incorporating regional and/or time-varying characteristics. Here, the space-time index $[\boldsymbol{s}][t]$ refers to the region covering $\boldsymbol{s}$ during the period containing t and thus spans a whole spatio-temporal grid on which the involved covariates are measured, e.g., district $\times$ month. We will later see that the endemic component therefore simply equals an inhomogeneous Poisson process for the event counts by cell of that grid.

The second, observation-driven epidemic component adds "infection pressure" from the set

$$I(\boldsymbol{s},t) = \{j : t_j < t \ \wedge \ t - t_j \leq \tau_j \ \wedge \ \|\boldsymbol{s} - \boldsymbol{s}_j\| \leq \delta_j\}$$

of past events and hence makes the process "self-exciting". During its infectious period of length τ_j and within its spatial interaction radius δ_j, the model assumes each event j to trigger further events, which are called offspring, secondary cases, or aftershocks, depending on the application. The triggering rate (or force of infection) is proportional to a log-linear predictor η_j associated with event-specific characteristics ("marks") $\boldsymbol{m}_j$, which are usually attached to the point pattern of events. The decay of infection pressure with increasing spatial and temporal distance from the infective event is modeled by parametric interaction functions f and g, respectively. A simple assumption for the time course of infectivity is $g(t) = 1$. Alternatives include exponential decay, a step function, or empirically derived functions such as Omori's law for aftershock intervals. With regard to spatial interaction, a Gaussian kernel $f(x) = \exp\{-x^2/(2\sigma^2)\}$ could be chosen. However, in modeling the spread of human infectious diseases on larger scales, a heavy-tailed power-law kernel $f(x) = (x+\sigma)^{-d}$ was found to perform better (Meyer and Held 2014). The (possibly infinite) upper bounds τ_j and δ_j provide a way of modeling event-specific interaction ranges. However, since these need to be pre-specified, a common assumption is $\tau_j \equiv \tau$ and $\delta_j \equiv \delta$, where the infectious period τ and the spatial interaction radius δ are determined by subject-matter considerations.

1.1. Model-based effective reproduction numbers

Similar to the simple SIR model (see, e.g., Keeling and Rohani 2008, Section 2.1), the above point process model (1) features a reproduction number derived from its branching process interpretation. As soon as an event occurs (individual becomes infected), it triggers offspring (secondary cases) around its origin $(\boldsymbol{s}_j, t_j)$ according to an inhomogeneous Poisson process with rate $\eta_j \, f(\|\boldsymbol{s} - \boldsymbol{s}_j\|) \, g(t - t_j)$. Since this triggering process is independent of the event's parentage and of other events, the expected number μ_j of events triggered by event j can be obtained by integrating the triggering rate over the observed interaction domain:

$$\mu_j = \eta_j \cdot \left[\int_0^{\min(T-t_j,\tau_j)} g(t) \, dt\right] \cdot \left[\int_{\boldsymbol{R}_j} f(\|\boldsymbol{s}\|) \, d\boldsymbol{s}\right] \, , \tag{2}$$

where

$$\boldsymbol{R}_j = (b(\boldsymbol{s}_j, \delta_j) \cap \boldsymbol{W}) - \boldsymbol{s}_j \tag{3}$$

is event j's influence region centered at $\boldsymbol{s}_j$, and $b(\boldsymbol{s}_j, \delta_j)$ denotes the disc centered at $\boldsymbol{s}_j$ with radius δ_j. Note that the above model-based reproduction number μ_j is event-specific since it depends on event marks through η_j, on the interaction ranges δ_j and τ_j, as well as on the event location $\boldsymbol{s}_j$ and time point t_j. If the model assumes unique interaction ranges δ and τ, a single reference number of secondary cases can be extrapolated from Equation 2 by imputing an unbounded domain $\boldsymbol{W} = \mathbb{R}^2$ and $T = \infty$ (Meyer, Warnke, Rössler, and Held 2016).

Equation 2 can also be motivated by looking at a spatio-temporal version of the simple SIR model wrapped into the `twinstim` class (1). This means: no endemic component, homogeneous force of infection ($\eta_j \equiv \beta$), homogeneous mixing in space ($f(x) = 1$, $\delta_j \equiv \infty$), and exponential decay of infectivity over time ($g(t) = e^{-\alpha t}$, $\tau_j \equiv \infty$). Then, for $T \to \infty$,

$$\mu = \beta \cdot \left[\int_0^\infty e^{-\alpha t}\, dt\right] \cdot \left[\int_{\boldsymbol{W}-\boldsymbol{s}_j} 1\, d\boldsymbol{s}\right] = \beta \cdot |\boldsymbol{W}|/\alpha\,,$$

which corresponds to the basic reproduction number known from the simple SIR model by interpreting $|\boldsymbol{W}|$ as the population size, β as the transmission rate and α as the removal rate. If $\mu < 1$, the process is sub-critical, i.e., its eventual extinction is almost sure.

However, it is crucial to understand that in a full model with an endemic component, new infections may always occur via "immigration". Hence, reproduction numbers in `twinstim` are adjusted for infections occurring independently of previous infections. This also means that a misspecified endemic component may distort model-based reproduction numbers (Meyer *et al.* 2016). Furthermore, under-reporting and implemented control measures imply that the estimates are to be thought of as *effective* reproduction numbers.

1.2. Likelihood inference

The log-likelihood of the point process model (1) is a function of all parameters in the log-linear predictors $\nu_{[\boldsymbol{s}][t]}$ and η_j and in the interaction functions f and g. It has the form

$$\left[\sum_{i=1}^{n} \log \lambda(\boldsymbol{s}_i, t_i)\right] - \int_0^T \int_{\boldsymbol{W}} \lambda(\boldsymbol{s}, t)\, \mathrm{d}\boldsymbol{s}\, \mathrm{d}t\,. \tag{4}$$

To estimate the model parameters, we maximize the above log-likelihood numerically using the quasi-Newton algorithm available through the R function `nlminb`. We thereby employ the analytical score function and an approximation of the expected Fisher information worked out by Meyer *et al.* (2012, Web Appendices A and B).

The space-time integral in the log-likelihood (4) poses no difficulties for the endemic component of $\lambda(\boldsymbol{s}, t)$, since $\nu_{[\boldsymbol{s}][t]}$ is defined on a spatio-temporal grid. However, integration of the epidemic component involves two-dimensional integrals $\int_{\boldsymbol{R}_i} f(\|\boldsymbol{s}\|)\, \mathrm{d}\boldsymbol{s}$ over the influence regions $\boldsymbol{R}_i$, which are represented by polygons (as is $\boldsymbol{W}$). Similar integrals appear in the score function, where $f(\|\boldsymbol{s}\|)$ is replaced by partial derivatives with respect to kernel parameters. Calculation of these integrals is trivial for (piecewise) constant f, but otherwise requires numerical integration. The R package **polyCub** (Meyer 2019) offers various cubature methods for polygonal domains. Of particular relevance for `twinstim` is the `polyCub.iso` method, which takes advantage of the assumed isotropy of spatial interaction such that numerical integration remains in only one dimension (Meyer and Held 2014, Supplement B, Section 2). We **memoise** (Wickham, Hester, Chang, Müller, and Cook 2021) the cubature function during log-likelihood maximization to avoid integration for unchanged parameters of f.

1.3. Special cases: Single-component models

If the *epidemic* component is omitted in Equation 1, the point process model becomes equivalent to a Poisson regression model for aggregated counts. This provides a link to ecological regression approaches in general and to the count data model `hhh4` illustrated in `vignette("hhh4")` and `vignette("hhh4_spacetime")`. To see this, recall that the endemic component $\nu_{[s][t]}$ is piecewise constant on the spatio-temporal grid with cells $([s],[t])$. Hence the log-likelihood (4) of an endemic-only `twinstim` simplifies to a sum over all these cells,

$$\sum_{[s],[t]} \left\{ Y_{[s][t]} \log \nu_{[s][t]} - |[s]|\,|[t]|\,\nu_{[s][t]} \right\} ,$$

where $Y_{[s][t]}$ is the aggregated number of events observed in cell $([s],[t])$, and $|[s]|$ and $|[t]|$ denote cell area and length, respectively. Except for an additive constant, the above log-likelihood is equivalently obtained from the Poisson model $Y_{[s][t]} \sim \text{Po}(|[s]|\,|[t]|\,\nu_{[s][t]})$. This relation offers a means of code validation using the established `glm` function to fit an endemic-only `twinstim` model – see the examples in `help("glm_epidataCS")`.

If, in contrast, the *endemic* component is omitted, all events are necessarily triggered by other observed events. For such a model to be identifiable, a prehistory of events must exist to trigger the first event, and interaction typically needs to be unbounded such that each event can actually be linked to potential source events.

1.4. Extension: `twinstim` with event types

To model the example data on invasive meningococcal disease in the remainder of this section, we actually need to use an extended version $\lambda(s,t,k)$ of Equation 1, which accounts for different event types k with own transmission dynamics. This introduces a further dimension in the point process, and the second log-likelihood component in Equation 4 accordingly splits into a sum over all event types. We refer to Meyer *et al.* (2012, Sections 2.4 and 3) for the technical details of this type-specific `twinstim` class. The basic idea is that the meningococcal finetypes share the same endemic pattern (e.g., seasonality), while infections of different finetypes are not associated via transmission. This means that the force of infection is restricted to previously infected individuals with the same bacterial finetype k, i.e., the epidemic sum in Equation 1 is over the set $I(s,t,k) = I(s,t) \cap \{j : k_j = k\}$. The implementation has limited support for type-dependent interaction functions f_{k_j} and g_{k_j} (not further considered here).

2. Data structure: `epidataCS`

The first step toward fitting a `twinstim` is to turn the relevant data into an object of the dedicated class `epidataCS`.[1] The primary ingredients of this class are a spatio-temporal point pattern (`events`) and its underlying observation region (`W`). An additional spatio-temporal grid (`stgrid`) holds (time-varying) area-level covariates for the endemic regression part. We exemplify this data class by the `epidataCS` object for the 636 cases of invasive meningococcal disease in Germany originally analyzed by Meyer *et al.* (2012). It is already contained in the **surveillance** package as `data("imdepi")` and has been constructed as follows:

[1] The suffix "CS" indicates that the data-generating point process is indexed in continuous space.

```
R> imdepi <- as.epidataCS(events = events, W = stateD, stgrid = stgrid,
+    qmatrix = diag(2), nCircle2Poly = 16)
```

The function `as.epidataCS` checks the consistency of the three data ingredients described in detail below. It also pre-computes auxiliary variables for model fitting, e.g., the individual influence regions (3), which are intersections of the observation region with discs approximated by polygons with `nCircle2Poly = 16` edges. The intersections are computed using functionality of the package **polyclip** (Johnson and Baddeley 2024). For multitype epidemics as in our example, the additional indicator matrix `qmatrix` specifies transmissibility across event types. An identity matrix corresponds to an independent spread of the event types, i.e., cases of one type can not produce cases of another type.

2.1. Data ingredients

The core `events` data must be provided in the form of a `SpatialPointsDataFrame` as defined by the package **sp** (Pebesma and Bivand 2024):

```
R> summary(events)

Object of class SpatialPointsDataFrame
Coordinates:
   min  max
x 4039 4665
y 2710 3525
Is projected: TRUE
proj4string :
[+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=km +no_defs]
Number of points: 636
Data attributes:
     time             tile    type       eps.t       eps.s        sex          agegrp
Min.   :   0.2  05354  : 34  B:336  Min.   :30  Min.   :200  female:292  [0,3)   :194
1st Qu.: 539.5  05370  : 27  C:300  1st Qu.:30  1st Qu.:200  male  :339  [3,19)  :279
Median :1155.0  11000  : 27         Median :30  Median :200  NA's  :  5  [19,Inf):162
Mean   :1192.7  05358  : 13         Mean   :30  Mean   :200              NA's    :  1
3rd Qu.:1808.0  05162  : 12         3rd Qu.:30  3rd Qu.:200
Max.   :2542.8  05382  : 12         Max.   :30  Max.   :200
                (Other):511
```

The associated event coordinates are residence postcode centroids, projected in the *European Terrestrial Reference System 1989* (in kilometer units) to enable Euclidean geometry. See the `spTransform`-methods for how to project latitude and longitude coordinates into a planar coordinate reference system (CRS). The data frame associated with these spatial coordinates ($\boldsymbol{s}_i$) contains a number of required variables and additional event marks (in the notation of Section 1: $\{(t_i, [\boldsymbol{s}_i], k_i, \tau_i, \delta_i, \boldsymbol{m}_i) : i = 1, \ldots, n\}$). For the IMD data, the event `time` is measured in days since the beginning of the observation period 2002–2008 and is subject to a tie-breaking procedure (described later). The `tile` column refers to the region of the spatio-temporal grid where the event occurred and here contains the official key of the administrative district of the patient's residence. There are two `type`s of events labeled as `"B"` and `"C"`, which refer to the serogroups of the two meningococcal finetypes *B:P1.7-2,4:F1-5* and *C:P1.5,2:F3-3* contained in the data. The `eps.t` and `eps.s` columns specify upper limits

for temporal and spatial interaction, respectively. Here, the infectious period is assumed to last a maximum of 30 days and spatial interaction is limited to a 200 km radius for all cases. The latter has numerical advantages for a Gaussian interaction function f with a relatively small standard deviation. For a power-law kernel, however, this restriction will be dropped to enable occasional long-range transmission. The last two data attributes displayed in the above `event` summary are covariates from the case reports: the gender and age group of the patient.

For the observation region `W`, we use a polygon representation of Germany's boundary. Since the observation region defines the integration domain in the point process log-likelihood (4), the more detailed the polygons of `W` are the longer it will take to fit a `twinstim`. It is thus advisable to sacrifice some shape details for speed by reducing the polygon complexity. In R this can be achieved via, e.g., `ms_simplify` from the **rmapshaper** package (Teucher and Russell 2020), or `simplify.owin` from **spatstat.geom** (Baddeley, Rubak, and Turner 2015). The **surveillance** package already contains a simplified representation of Germany's boundaries:

```
R> load(system.file("shapes", "districtsD.RData", package = "surveillance"))
```

This file contains both the `SpatialPolygonsDataFrame` `districtsD` of Germany's 413 administrative districts as at January 1, 2009, as well as their union `stateD`. These boundaries are projected in the same CRS as the `events` data.

The `stgrid` input for the endemic model component is a data frame with (time-varying) area-level covariates, e.g., socio-economic or ecological characteristics. In our example:

```
      start stop  tile   area popdensity
1         0   31 01001   56.4     1557.1
2         0   31 01002  118.7     1996.6
3         0   31 01003  214.2      987.6
...     ...  ...   ...    ...        ...
34690  2526 2557 16075 1148.5       79.2
34691  2526 2557 16076  843.5      133.6
34692  2526 2557 16077  569.1      181.5
```

Numeric (`start`,`stop`] columns index the time periods and the factor variable `tile` identifies the regions of the grid. Note that the given time intervals (here: months) also define the resolution of possible time trends and seasonality of the piecewise constant endemic intensity. We choose monthly intervals to reduce package size and computational cost compared to the weekly resolution originally used by Meyer *et al.* (2012) and Meyer and Held (2014). The above `stgrid` data frame thus consists of 7 (years) times 12 (months) blocks of 413 (districts) rows each. The `area` column gives the area of the respective `tile` in square kilometers (compatible with the CRS used for `events` and `W`). A geographic representation of the regions in `stgrid` is not required for model estimation, and is thus not part of the `epidataCS` class. In our example, the area-level data only consists of the population density `popdensity`, whereas Meyer *et al.* (2012) additionally incorporated (lagged) weekly influenza counts by district as a time-dependent covariate.

2.2. Data handling and visualization

The generated `epidataCS` object `imdepi` is a simple list of the checked ingredients `events`, `stgrid`, `W`, `qmatrix`. Several methods for data handling and visualization are available for such objects as listed in Table 1 and briefly presented in the remainder of this section.

Display	Subset	Extract	Modify	Convert	Other
`print`	`[`	`nobs`	`update`	`as.epidata`	`coerce`
`summary`	`head`	`marks`	`untie`	*`epidataCS2sts`*	`initialize`
`plot`	`tail`	*`getSourceDists`*			`show`
`animate`	`subset`				`slotsFromS3`
`as.stepfun`					

Table 1: Generic and *non-generic* functions applicable to `epidataCS` objects.

Printing an `epidataCS` object presents some metadata and the first 6 events by default:

```
R> imdepi

Observation period: 0 - 2557
Observation window (bounding box): [4031, 4672] x [2684, 3550]
Spatio-temporal grid (not shown): 84 time blocks x 413 tiles
Types of events: "B" "C"
Overall number of events: 636

   coordinates    time  tile type eps.t eps.s    sex   agegrp BLOCK start popdensity
1 (4112, 3203)  0.2117 05554    B    30   200   male   [3,19)     1     0       260.9
2 (4123, 3077)  0.7124 05382    C    30   200   male   [3,19)     1     0       519.4
3 (4412, 2916)  5.5910 09574    B    30   200 female [19,Inf)     1     0       209.4
4 (4203, 2880)  7.1170 08212    B    30   200 female   [3,19)     1     0      1665.6
5 (4128, 3223) 22.0595 05554    C    30   200   male   [3,19)     1     0       260.9
6 (4090, 3178) 24.9544 05170    C    30   200   male   [3,19)     1     0       454.7
[....]
```

During conversion to `epidataCS`, the last three columns `BLOCK` (time interval index), `start` and `popdensity` have been merged from the checked `stgrid` to the `events` data frame. The event marks including time and location can be extracted in a standard data frame by `marks(imdepi)` – inspired by package **spatstat** – and this is summarized by `summary(imdepi)`.

The number of potential sources of infection per event (denoted `|.sources|` in the above output) is additionally summarized. It is determined by the events' maximum ranges of interaction `eps.t` and `eps.s`. The event-specific set of potential sources is stored in the (hidden) list `imdepi$events$.sources` (events are referenced by row index), and the event-specific numbers of potential sources are stored in the summarized object as `simdepi$nSources`.

A simple plot of the number of infectives as a function of time (Figure 1) can be obtained by the step function converter:

```
R> plot(as.stepfun(imdepi), xlim = summary(imdepi)$timeRange, xaxs = "i",
+    xlab = "Time [days]", ylab = "Current number of infectives", main = "")
```

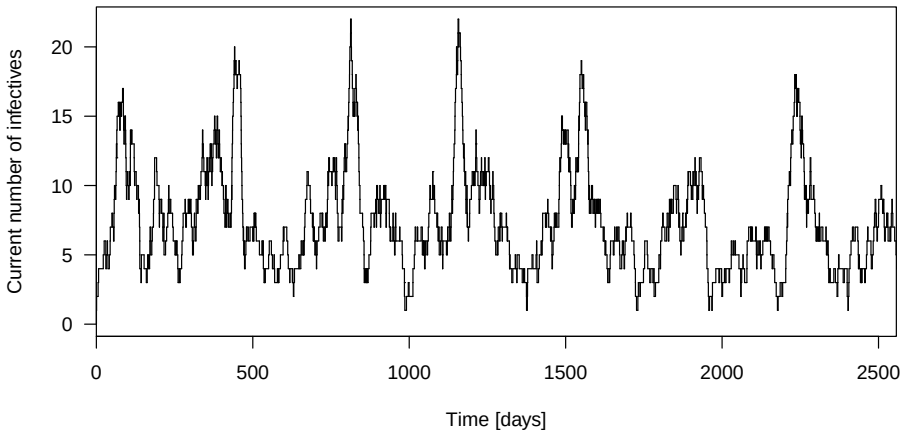
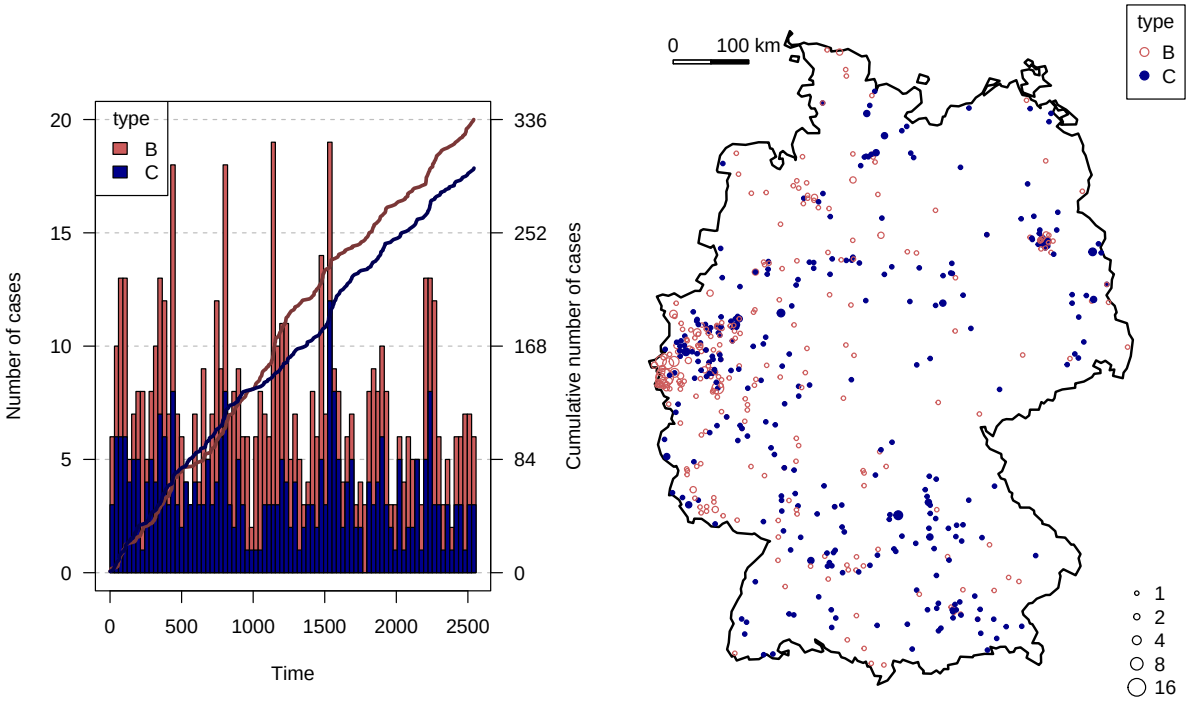


Figure 1: Time course of the number of infectives assuming infectious periods of 30 days.

The `plot`-method for `epidataCS` offers aggregation of the events over time or space:

```
R> plot(imdepi, "time", col = c("indianred", "darkblue"), ylim = c(0, 20))
R> plot(imdepi, "space", lwd = 2,
+    points.args = list(pch = c(1, 19), col = c("indianred", "darkblue")))
R> layout.scalebar(imdepi$W, scale = 100, labels = c("0", "100 km"), plot = TRUE)
```



(a) Temporal pattern.

(b) Spatial pattern.

Figure 2: Occurrence of the two finetypes viewed in the temporal and spatial dimensions.

The time-series plot (Figure 2a) shows the monthly aggregated number of cases by finetype in a stacked histogram as well as each type's cumulative number over time. The spatial plot (Figure 2b) shows the observation window `W` with the locations of all cases (by type), where the areas of the points are proportional to the number of cases at the respective location. Additional shading by the population is possible and exemplified in `help("plot.epidataCS")`. An animation may provide additional insight and can be produced by the corresponding `animate`-method. For instance, to look at the first year of the B-type in a weekly sequence of snapshots in a web browser (using facilities of the **animation** package of Xie 2013):

```
R> animation::saveHTML(
+    animate(subset(imdepi, type == "B"), interval = c(0, 365), time.spacing = 7),
+    nmax = Inf, interval = 0.2, loop = FALSE, title = "First year of type B")
```

Selecting events from `epidataCS` as for the animation above is enabled by the `[`- and `subset`-methods, which return a new `epidataCS` object containing only the selected `events`.

A limited data sampling resolution may lead to tied event times or locations, which are in conflict with a continuous spatio-temporal point process model. For instance, a temporal residual analysis would suggest model deficiencies (Meyer *et al.* 2012, Figure 4), and a power-law kernel for spatial interaction may diverge if there are events with zero distance to potential source events (Meyer and Held 2014). The function `untie` breaks ties by random shifts. This has already been applied to the event *times* in the provided `imdepi` data by subtracting a $U(0, 1)$-distributed random number from the original dates. The event *coordinates* in the IMD data are subject to interval censoring at the level of Germany's postcode regions. A possible replacement for the given centroids would thus be a random location within the corresponding postcode area. Lacking a suitable shapefile, Meyer and Held (2014) shifted all locations by a random vector with length up to half the observed minimum spatial separation:

```
R> eventDists <- dist(coordinates(imdepi$events))
R> minsep <- min(eventDists[eventDists > 0])
R> set.seed(321)
R> imdepi_untied <- untie(imdepi, amount = list(s = minsep / 2))
```

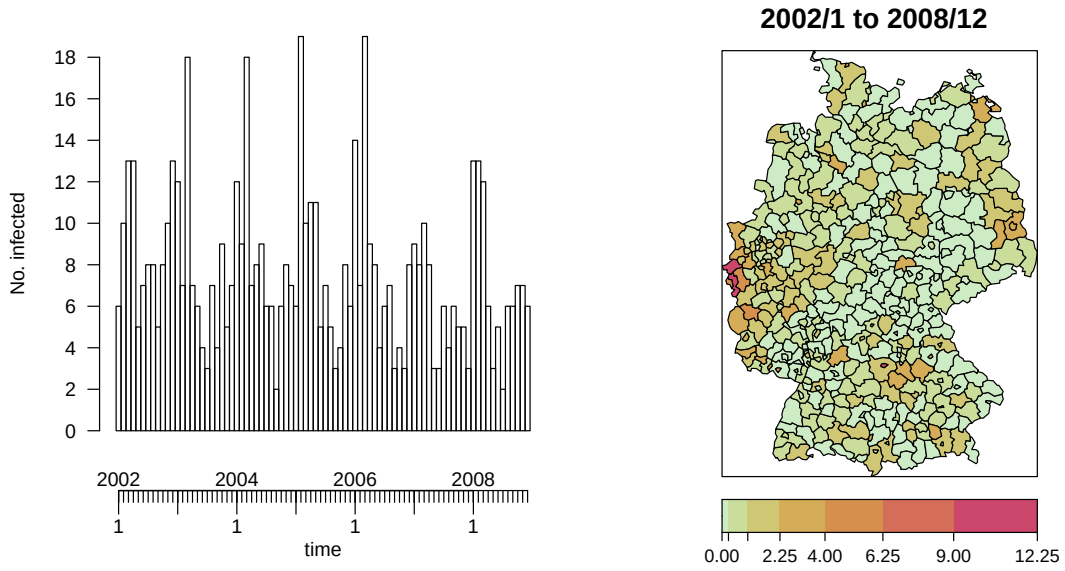
Note that random tie-breaking requires sensitivity analyses as discussed by Meyer and Held (2014), but these are skipped here for the sake of brevity.

The `update`-method is useful to change the values of the maximum interaction ranges `eps.t` and `eps.s`, since it takes care of the necessary updates of the hidden auxiliary variables in an `epidataCS` object. For unbounded spatial interaction:

```
R> imdepi_untied_infeps <- update(imdepi_untied, eps.s = Inf)
```

Last but not least, `epidataCS` can be aggregated to `epidata` (from `vignette("twinSIR")`) or `sts` (from `vignette("hhh4_spacetime")`). The method `as.epidata.epidataCS` aggregates events by region (`tile`), and the function `epidataCS2sts` yields counts by region and time interval. The latter could be analyzed by an areal time-series model such as `hhh4` (see `vignette("hhh4_spacetime")`). We can also use `sts` visualizations, e.g. (Figure 3):

```
R> imdsts <- epidataCS2sts(imdepi, freq = 12, start = c(2002, 1), tiles = districtsD,
+                          neighbourhood = NULL) # skip adjacency matrix (needs spdep)
R> plot(imdsts, type = observed ~ time)
R> plot(imdsts, type = observed ~ unit, population = districtsD$POPULATION / 100000)
```



(a) Time series of monthly counts.

(b) Disease incidence (per 100 000 inhabitants).

Figure 3: IMD cases (joint types) aggregated as an `sts` object by month and district.

3. Modeling and inference

Having prepared the data as an object of class `epidataCS`, the function `twinstim` can be used to perform likelihood inference for conditional intensity models of the form (1). The main arguments for `twinstim` are the formulae of the `endemic` and `epidemic` linear predictors ($\nu_{[s][t]} = \exp(\texttt{endemic})$ and $\eta_j = \exp(\texttt{epidemic})$), and the spatial and temporal interaction functions `siaf` (f) and `tiaf` (g), respectively. Both formulae are parsed internally using the standard `model.frame` toolbox from package **stats** and thus can handle factor variables and interaction terms. While the `endemic` linear predictor incorporates covariates from `stgrid`, the `epidemic` formula may use both `stgrid` variables and event marks to be associated with the force of infection. For the interaction functions, several alternatives are predefined as listed in Table 2. They are applicable out-of-the-box and illustrated as part of the following modeling exercise for the IMD data. Own interaction functions can also be implemented following the structure described in `help("siaf")` and `help("tiaf")`, respectively.

Spatial (`siaf.*`)	Temporal (`tiaf.*`)
`constant`	`constant`
`exponential`	`exponential`
`gaussian`	`step`
`powerlaw`	
`powerlaw1`	
`powerlawL`	
`step`	
`student`	

Table 2: Predefined spatial and temporal interaction functions.

3.1. Basic example

To illustrate statistical inference with `twinstim`, we will estimate several models for the simplified and "untied" IMD data presented in Section 2. In the endemic component, we include the district-specific population density as a multiplicative offset, a (centered) time trend, and a sinusoidal wave of frequency $2\pi/365$ to capture seasonality, where the `start` variable from `stgrid` measures time:

```
R> (endemic <- addSeason2formula(~offset(log(popdensity)) + I(start / 365 - 3.5),
+    period = 365, timevar = "start"))

~offset(log(popdensity)) + I(start/365 - 3.5) + sin(2 * pi *
    start/365) + cos(2 * pi * start/365)
```

See Held and Paul (2012, Section 2.2) for how such sine/cosine terms reflect seasonality. Because of the aforementioned integrations in the log-likelihood (4), it is advisable to first fit an endemic-only model to obtain reasonable start values for more complex epidemic models:

```
R> imdfit_endemic <- twinstim(endemic = endemic, epidemic = ~0,
+    data = imdepi_untied, subset = !is.na(agegrp))
```

We exclude the single case with unknown age group from this analysis since we will later estimate an effect of the age group on the force of infection.

Many of the standard functions to access model fits in R are also implemented for `twinstim` fits (see Table 3). For example, we can produce the usual model summary:

```
R> summary(imdfit_endemic)

Call:
twinstim(endemic = endemic, epidemic = ~0, data = imdepi_untied,
    subset = !is.na(agegrp))

Coefficients of the endemic component:
                          Estimate Std. Error z value Pr(>|z|)
h.(Intercept)             -20.3683     0.0419 -486.24  < 2e-16 ***
h.I(start/365 - 3.5)       -0.0444     0.0200   -2.22    0.027 *
h.sin(2 * pi * start/365)   0.2733     0.0576    4.75  2.0e-06 ***
h.cos(2 * pi * start/365)   0.3509     0.0581    6.04  1.5e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

No epidemic component.

AIC:  19166
Log-likelihood: -9579
```

Because of the aforementioned equivalence of the endemic component with a Poisson regression model, the coefficients can be interpreted as log rate ratios in the usual way. For instance, the endemic rate is estimated to decrease by `1 - exp(coef(imdfit_endemic)[2])` $= 4.3\%$ per year. Coefficient correlations can be retrieved via the argument `correlation = TRUE` in the `summary` call just like for `summary.glm`, or via `cov2cor(vcov(imdfit_endemic))`.

Display	Extract	Modify	Other
`print`	`nobs`	`update`	`simulate`
`summary`	`vcov`	`add1`	`all.equal`
`xtable`	`coeflist`	`drop1`	*`epitest`*
`plot`	`logLik`	*`stepComponent`*	
`intensityplot`	`extractAIC`		
`iafplot`	`profile`		
`checkResidualProcess`	`residuals`		
	`terms`		
	`R0`		
	`intensity.twinstim`		
	`simpleR0`		

Table 3: Generic and *non-generic* functions applicable to `twinstim` objects. Note that there is no need for specific `coef`, `confint`, `AIC` or `BIC` methods, since the respective default methods from package **stats** apply outright.

We now update the endemic model to take additional spatio-temporal dependence between events into account. Infectivity shall depend on the meningococcal finetype and the age group of the patient, and is assumed to be constant over time (default), $g(t) = \mathbb{I}_{(0,30]}(t)$, with a Gaussian distance-decay $f(x) = \exp\{-x^2/(2\sigma^2)\}$. This model was originally selected by Meyer *et al.* (2012) and can be fitted as follows:

```
R> imdfit_Gaussian <- update(imdfit_endemic, epidemic = ~type + agegrp,
+    siaf = siaf.gaussian(), cores = 2 * (.Platform$OS.type == "unix"))
```

On Unix-alikes, the numerical integrations of $f(\|\boldsymbol{s}\|)$ in the log-likelihood and $\frac{\partial f(\|\boldsymbol{s}\|)}{\partial \log \sigma}$ in the score function (note that σ is estimated on the log-scale) can be performed in parallel via `mclapply` *et al.* from the base package **parallel**, here with `cores = 2` processes.

Table 4 shows the output of `twinstim`'s `xtable` method (Dahl, Scott, Roosen, Magnusson, and Swinton 2019) applied to the above model fit, providing a table of estimated rate ratios for the endemic and epidemic effects. The alternative `toLatex` method simply translates the `summary` table of coefficients to LaTeX without `exp`-transformation. On the subject-matter level, we can conclude from Table 4 that the meningococcal finetype of serogroup C is less than half as infectious as the B-type, and that patients in the age group 3 to 18 years are estimated to cause twice as many secondary infections as infants aged 0 to 2 years.

	RR	95% CI	p-value
`h.I(start/365 - 3.5)`	0.955	0.91–1.00	0.039
`h.sin(2 * pi * start/365)`	1.243	1.09–1.41	0.0008
`h.cos(2 * pi * start/365)`	1.375	1.21–1.56	<0.0001
`e.typeC`	0.402	0.24–0.68	0.0007
`e.agegrp[3,19)`	2.022	1.07–3.83	0.031
`e.agegrp[19,Inf)`	0.787	0.32–1.94	0.60

Table 4: Estimated rate ratios (RR) and associated Wald confidence intervals (CI) for endemic (`h.`) and epidemic (`e.`) terms. This table was generated by `xtable(imdfit_Gaussian)`.

3.2. Model-based effective reproduction numbers

The event-specific reproduction numbers (2) can be extracted from fitted `twinstim` objects via the `R0` method. For the above IMD model, we obtain the following mean numbers of secondary infections by finetype:

```
R> R0_events <- R0(imdfit_Gaussian)
R> tapply(R0_events, marks(imdepi_untied)[names(R0_events), "type"], mean)

      B       C
0.21564 0.09542
```

Confidence intervals can be obtained via Monte Carlo simulation, where Equation 2 is repeatedly evaluated with parameters sampled from the asymptotic multivariate normal distribution of the maximum likelihood estimate. For this purpose, the `R0`-method takes an argument `newcoef`, which is exemplified in `help("R0")`.

3.3. Interaction functions

Figure 4 shows several estimated spatial interaction functions, which can be plotted by, e.g., `plot(imdfit_Gaussian, "siaf")`.

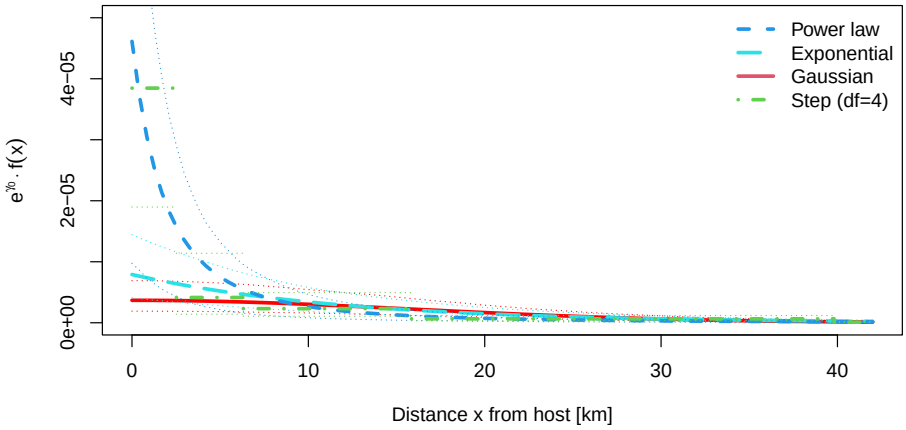


Figure 4: Various estimates of spatial interaction (scaled by the epidemic intercept γ_0).

The estimated standard deviation $\hat{\sigma}$ of the Gaussian kernel is:

```
R> exp(cbind("Estimate" = coef(imdfit_Gaussian)["e.siaf.1"],
+            confint(imdfit_Gaussian, parm = "e.siaf.1")))

         Estimate 2.5 % 97.5 %
e.siaf.1    15.92 13.58  18.65
```

Meyer and Held (2014) found that a power-law decay of spatial interaction more appropriately describes the spread of human infectious diseases. A power-law kernel concentrates on short-range interaction, but also exhibits a heavier tail reflecting occasional transmission over large distances. To estimate the power law $f(x) = (x + \sigma)^{-d}$, we use the prepared `eps.s = Inf` version of the `epidataCS` object, and update the model as follows:

```
R> imdfit_powerlaw <- update(imdfit_Gaussian, siaf = siaf.powerlaw(),
+    data = imdepi_untied_infeps,
+    start = c("e.(Intercept)" = -6.2, "e.siaf.1" = 1.5, "e.siaf.2" = 0.9))
```

To reduce the runtime of this example, we specified convenient `start` values for some parameters. The estimated parameters $(\hat{\sigma}, \hat{d})$ are:

```
R> exp(cbind("Estimate" = coef(imdfit_powerlaw)[c("e.siaf.1", "e.siaf.2")],
+            confint(imdfit_powerlaw, parm = c("e.siaf.1", "e.siaf.2"))))
```

```
         Estimate 2.5 % 97.5 %
e.siaf.1    4.660 1.825  11.90
e.siaf.2    2.491 1.809   3.43
```

Sometimes σ is difficult to estimate, and also in this example, its confidence interval is relatively large. The one-parameter version `siaf.powerlaw1` can be used to estimate a power-law decay with fixed $\sigma = 1$. A more common option is the exponential kernel $f(x) = \exp(-x/\sigma)$:

```
R> imdfit_exponential <- update(imdfit_Gaussian, siaf = siaf.exponential())
```

Table 2 also lists the step function kernel as an alternative, which is particularly useful for two reasons. First, it is a more flexible approach since it estimates interaction between the given knots without assuming an overall functional form. Second, the spatial integrals in the log-likelihood can be computed analytically for the step function kernel, which therefore offers a quick estimate of spatial interaction. We update the Gaussian model to use four steps at log-equidistant knots up to an interaction range of 100 km:

```
R> imdfit_step4 <- update(imdfit_Gaussian,
+    siaf = siaf.step(exp(1:4 * log(100) / 5), maxRange = 100))
```

Figure 4 suggests that the estimated step function is in line with the power law. Note that suitable knots for the step function could also be derived from quantiles of the observed distances between events and their potential source events, e.g.:

```
R> quantile(getSourceDists(imdepi_untied_infeps, "space"), c(1,2,4,8)/100)
```

```
    1%     2%     4%     8%
 5.742 10.352 19.334 35.957
```

For the temporal interaction function $g(t)$, model updates and plots are similarly possible, e.g., using `update(imdfit_Gaussian, tiaf = tiaf.exponential())`. However, the events in the IMD data are too rare to infer the time-course of infectivity with confidence.

3.4. Model selection

```
R> AIC(imdfit_endemic, imdfit_Gaussian, imdfit_exponential, imdfit_powerlaw, imdfit_step4)
```

```
                   df   AIC
imdfit_endemic      4 19166
imdfit_Gaussian     9 18967
imdfit_exponential  9 18949
imdfit_powerlaw    10 18940
imdfit_step4       12 18933
```

Akaike's Information Criterion (AIC) suggests superiority of the power-law vs. the exponential, Gaussian, and endemic-only models. The more flexible step function yields the best AIC value, but its shape strongly depends on the chosen knots and is not guaranteed to be monotonically decreasing. The function `stepComponent` – a wrapper around the `step` function from **stats** – can be used to perform AIC-based stepwise selection within a given model component.

3.5. Model diagnostics

The element `"fittedComponents"` of a `twinstim` object contains the endemic and epidemic values of the estimated intensity at each event occurrence. However, plots of the conditional intensity (and its components) as a function of location or time provide more insight into the fitted process. Evaluation of `intensity.twinstim` requires the model environment to be stored with the fit. By default, `model = FALSE` in `twinstim`, but if the data are still available, the model environment can also be added afterwards using the convenient `update` method:

```
R> imdfit_powerlaw <- update(imdfit_powerlaw, model = TRUE)
```

Figure 5 shows an `intensityplot` of the fitted "ground" intensity $\sum_{k=1}^{2} \int_{\boldsymbol{W}} \hat{\lambda}(\boldsymbol{s}, t, k)\, \mathrm{d}\boldsymbol{s}$:

```
R> intensityplot(imdfit_powerlaw, which = "total", aggregate = "time", types = 1:2)
```

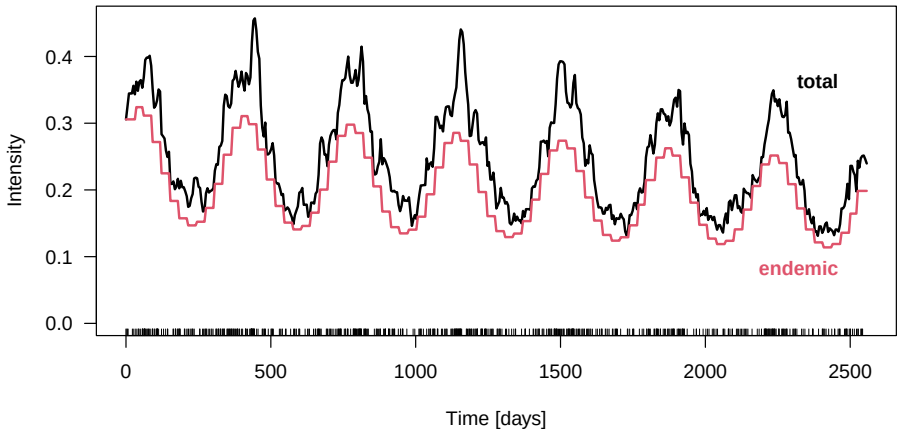


Figure 5: Fitted "ground" intensity process aggregated over space and both types.

The estimated endemic intensity component has also been added to the plot. It exhibits strong seasonality and a slow negative trend. The proportion of the endemic intensity is rather constant along time since no major outbreaks occurred. This proportion can be visualized separately by specifying `which = "endemic proportion"` in the above call.

Spatial `intensityplots` as in Figure 6 can be produced via `aggregate = "space"` and require a geographic representation of `stgrid`. The epidemic proportion is naturally high around clusters of cases and even more so if the population density is low.

Another diagnostic tool is the function `checkResidualProcess` (Figure 7), which transforms the temporal "residual process" in such a way that it exhibits a uniform distribution and lacks serial correlation if the fitted model describes the true CIF well (see Ogata 1988, Section 3.3).

4. Simulation

To identify regions with unexpected IMD dynamics, Meyer *et al.* (2012) compared the observed numbers of cases by district to the respective 2.5% and 97.5% quantiles of 100 simulations from the selected model. Furthermore, simulations allow us to investigate the stochastic volatility of the endemic-epidemic process, to obtain probabilistic forecasts, and to perform parametric bootstrap of the spatio-temporal point pattern.

The simulation algorithm we apply is described in Meyer *et al.* (2012, Section 4). It requires a geographic representation of the `stgrid`, as well as functionality for sampling locations from the spatial kernel $f_2(\boldsymbol{s}) := f(\|\boldsymbol{s}\|)$. This is implemented for all predefined spatial interaction functions listed in Table 2. Event marks are by default sampled from their respective empirical distribution in the original data. The following code runs *a single* simulation over the last year based on the estimated power-law model:

```
R> imdsim <- simulate(imdfit_powerlaw, nsim = 1, seed = 1, t0 = 2191, T = 2555,
+    data = imdepi_untied_infeps, tiles = districtsD)
```

This yields an object of the class `simEpidataCS`, which extends `epidataCS`. It carries additional components from the generating model to enable an `R0`-method and `intensityplots` for simulated data. Figure 8 shows the cumulative number of cases from the simulation appended to the first six years of data.

A special feature of such simulated epidemics is that the source of each event is known:

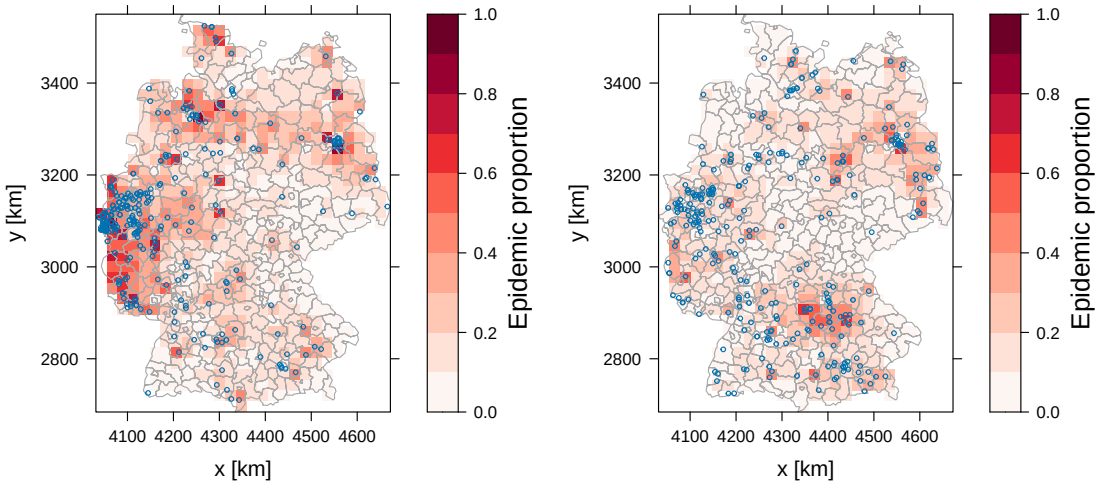
```
R> table(imdsim$events$source > 0, exclude = NULL)

FALSE  TRUE  <NA>
   54    21     3
```

The stored `source` value is 0 for endemic events, `NA` for events of the prehistory but still infective at `t0`, and otherwise corresponds to the row index of the infective source.

References

Baddeley A, Rubak E, Turner R (2015). *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC, London. ISBN 978-1-4822-1020-0.

Dahl DB, Scott D, Roosen C, Magnusson A, Swinton J (2019). *xtable: Export Tables to LaTeX or HTML*. R package version 1.8-4, URL http://xtable.r-forge.r-project.org/.



(a) Type B. (b) Type C.

Figure 6: Epidemic proportion of the fitted intensity process accumulated over time by type.

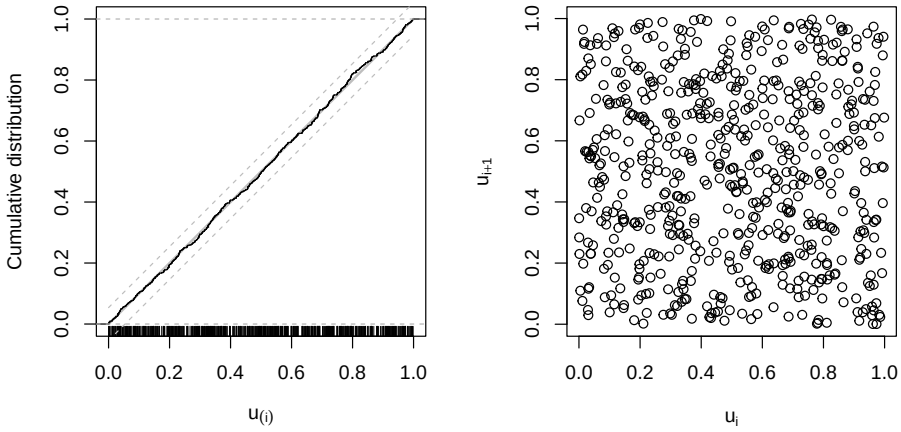


Figure 7: `checkResidualProcess(imdfit_powerlaw)`. The left-hand plot shows the `ecdf` of the transformed residuals with a 95% confidence band obtained by inverting the corresponding Kolmogorov-Smirnov test (no evidence for deviation from uniformity). The right-hand plot suggests absence of serial correlation.

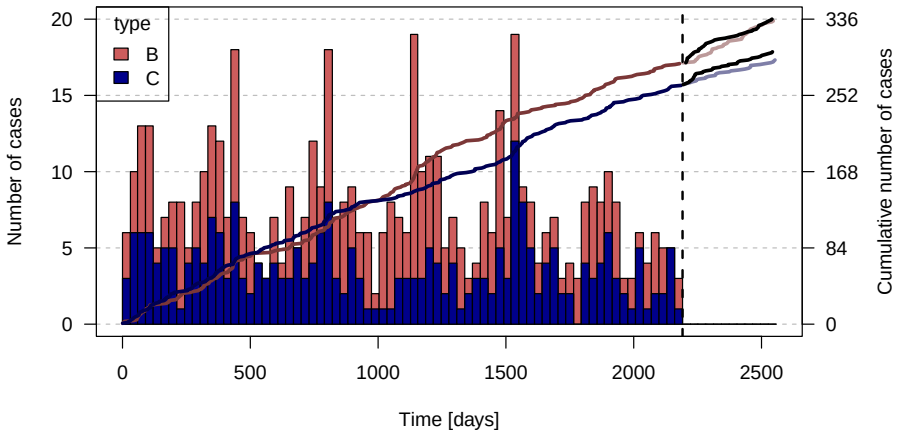


Figure 8: Simulation-based forecast of the cumulative number of cases by finetype in the last two years. The black lines correspond to the observed numbers.

Daley DJ, Vere-Jones D (2003). *An Introduction to the Theory of Point Processes*, volume I: Elementary Theory and Methods of *Probability and its Applications*. 2nd edition. Springer-Verlag, New York. ISBN 0-387-95541-0.

Held L, Paul M (2012). "Modeling seasonality in space-time infectious disease surveillance data." *Biometrical Journal*, **54**(6), 824–843. doi:10.1002/bimj.201200037.

Johnson A, Baddeley A (2024). *polyclip: Polygon Clipping*. R package version 1.10-7, URL https://www.angusj.com.

Keeling MJ, Rohani P (2008). *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press.

Meyer S (2019). "polyCub: An R package for integration over polygons." *Journal of Open Source Software*, **4**(34), 1056. doi:10.21105/joss.01056.

Meyer S, Elias J, Höhle M (2012). "A space-time conditional intensity model for invasive meningococcal disease occurrence." *Biometrics*, **68**(2), 607–616. doi:10.1111/j.1541-0420.2011.01684.x.

Meyer S, Held L (2014). "Power-law models for infectious disease spread." *Annals of Applied Statistics*, **8**(3), 1612–1639. doi:10.1214/14-AOAS743.

Meyer S, Held L, Höhle M (2017). "Spatio-temporal analysis of epidemic phenomena using the R package surveillance." *Journal of Statistical Software*, **77**(11), 1–55. doi:10.18637/jss.v077.i11.

Meyer S, Warnke I, Rössler W, Held L (2016). "Model-based testing for space-time interaction using point processes: An application to psychiatric hospital admissions in an urban area." *Spatial and Spatio-temporal Epidemiology*, **17**, 15–25. doi:10.1016/j.sste.2016.03.002.

Ogata Y (1988). "Statistical models for earthquake occurrences and residual analysis for point processes." *Journal of the American Statistical Association*, **83**(401), 9–27.

Pebesma E, Bivand R (2024). *sp: Classes and Methods for Spatial Data*. R package version 2.1-4, URL https://github.com/edzer/sp/.

Teucher A, Russell K (2020). *rmapshaper: Client for 'mapshaper' for 'Geospatial' Operations*. URL https://CRAN.R-project.org/package=rmapshaper.

Wickham H, Hester J, Chang W, Müller K, Cook D (2021). *memoise: 'Memoisation' of Functions*. R package version 2.0.1, URL https://memoise.r-lib.org.

Xie Y (2013). "animation: An R Package for Creating Animations and Demonstrating Statistical Methods." *Journal of Statistical Software*, **53**(1), 1–27. doi:10.18637/jss.v053.i01.