SpTe2M: An R Package for Nonparametric Modeling and Monitoring of Spatio-Temporal Data

Kai Yang[1] and Peihua Qiu[2]

[1]Division of Biostatistics, Medical College of Wisconsin, Milwaukee, WI 53226

[2]Department of Biostatistics, University of Florida, Gainesville, FL 32610

Abstract

Spatio-temporal data are common in practice. Such data often have complicated structures that are difficult to describe by parametric statistical models. Thus, it is often challenging to analyze spatio-temporal data effectively since most existing statistical methods and software packages in the literature are based on parametric modeling and cannot handle certain applications properly. This paper introduces the new R package **SpTe2M**, which is developed for implementing some recent nonparametric methods for modeling and monitoring spatio-temporal data. This package provides analytic tools for modeling spatio-temporal data nonparametrically and for monitoring dynamic spatial processes sequentially over time. It can be used for different applications, including disease surveillance, environmental monitoring, and more. The use of the package is demonstrated using the Florida influenza-like illness data observed during 2012-2014 and the PM2.5 concentration data in China collected during 2014-2016.

Keywords: Change detection; Covariance estimation; Disease surveillance; Local smoothing; Spatio-temporal data; Statistical process control.

1 Introduction

Spatio-temporal data become increasingly popular in many disciplines, including agriculture, climate science, epidemiology, neuroscience, social science, and more. In the literature, there have been many existing methods developed for analyzing spatio-temporal data (e.g., Cressie and Wikle 2011, Diggle et al. 2013). However, spatio-temporal data often have complicated structures, including complex spatio-temporal variation, latent spatio-temporal correlation, and unknown data distribution, which can hardly be described properly by parametric models (cf., Yang and Qiu 2020). But most existing methods for analyzing spatio-temporal data are parametric, and it is well demonstrated in the literature that nonparametric methods often perform better than the parametric methods in cases when some of the parametric assumptions are violated (cf., Subection 3.2, Qiu and Yang 2023). Therefore, there is a need to develop nonparametric methods and the corresponding software packages for analyzing spatio-temporal data. This paper introduces a new R package, called **SpTe2M**, that provides analytic tools for modeling spatio-temporal data nonparametrically and for monitoring dynamic spatial processes sequentially over time.

In the literature, there are some statistical methods and software packages available for analyzing spatio-temporal data. One popular approach is the dynamic spatio-temporal modeling framework that was first discussed in Stroud et al. (2001) and can be implemented by using the R package **spBayes** (Finley et al. 2015). In the dynamic spatio-temporal model, spatial observations are assumed to be normally distributed, and the sequence of observed spatial response surfaces over time is assumed to follow a linear evolution process. Some of its generalized versions have been proposed to extend the original dynamic spatio-temporal model to cases with multiple responses (Bradley et al. 2015) and/or cases with a nonlinear evolution equation (Wikle and Hooten 2010). As an alternative, Møller et al. (1998) proposed a framework for modeling a spatio-temporal point process based on the log-Gaussian Cox process, where the logarithm of the related spatio-temporal intensity function is assumed to be a spatio-temporal Gaussian process, conditional on which the observed spatio-temporal data are assumed to have a Poisson distribution (e.g., Diggle et al. 2013). This method can be implemented using the **lgcp** package (Taylor et al. 2013). There are some other methods for spatio-temporal modeling and prediction, including the hierarchical Bayesian spatio-temporal modeling method via the R package **spTimer** (Bakar and Sahu 2015), the lattice kriging method via the R package **LatticeKrig** (Nychka et al. 2015), the local approximate Gaussian process method via the R package **laGP** (Gramacy 2016), the fixed rank kriging method via the R package **FRK** (Zammit-Mangion and Cressie 2017), the metakriging method (Guhaniyogi and Banerjee 2018), the non-stationary Gaussian process method via the R package **BayesNSGP** (Risser and Turek 2020), and the nearest neighbor Gaussian process method via the R package **spNNGP** (Finley et al. 2020).

The existing methods mentioned above require various parametric assumptions on the spatio-temporal data variation, spatio-temporal data correlation, and/or data distribution. Such parametric assumptions are rarely valid in practice, and nonparametric methods usually perform better in such cases. To fill the gap, Yang and Qiu (2018, 2019, 2022) developed a nonparametric framework for modeling spatio-temporal data based on local kernel smoothing, in which no parametric assumptions were imposed on the spatio-temporal mean function, spatio-temporal covariance function, and data distribution. Thus, this spatio-temporal data modeling approach is flexible and can be used reliably in many applications. Based on this modeling approach, Yang and Qiu (2020) and Qiu and Yang (2021, 2023) proposed several methods for online monitoring of spatial processes and detecting possible shifts in their distributions. These spatial process monitoring methods can be used to solve many real-world applications, including environmental monitoring, disease surveillance, and more.

Regarding spatial process monitoring, major existing methods and software packages in the literature include those based on the scan statistics via the R package **scanstatistics** (Allévius 2018) and those based on the Knox statistics via the R package **surveillance** (Meyer et al. 2017). The scan and Knox methods are designed mainly for retrospective data analysis, in which the time interval is fixed in advance. They could not accommodate some common structures in the observed spatio-temporal data, including seasonality and spatio-temporal data correlation. They require the data distribution to have a parametric form (e.g., normal, Poisson, or negative binomial). As a comparison, the online process monitoring methods discussed in Yang and Qiu (2020) and Qiu and Yang (2021, 2023) do not require such parametric assumptions and can accommodate complex structures in the observed spatio-temporal data.

To help researchers to use the nonparametric spatio-temporal data modeling and monitoring methods suggested in Yang and Qiu (2018, 2019, 2020, 2022) and Qiu and Yang (2021, 2023), the R package **SpTe2M** has been developed to implement these methods. This paper provides an introduction and some demonstrations of the package. The remainder of the paper is organized as follows. Section 2 provides a brief overview of the methods that can be implemented in the **SpTe2M** package. Section 3 demonstrates the functionality of the package using a real-world dataset about the influenza-like illness in Florida observed during 2012-2014. Section 4 discusses the application of the package to another real data example about the PM2.5 concentrations in China during 2014-2016. Finally, Section 5 provides some concluding remarks.

2 Overview of the Methods Implemented in SpTe2M

2.1 Nonparametric modeling of spatio-temporal data

Nonparametric spatio-temporal model. Let $\{y(t_i, \boldsymbol{s}_{ij}), j = 1, \ldots, m_i, i = 1, \ldots, n\}$ be the observed spatio-temporal data, where $t_i \in [0, T]$ is the ith observation time, $\boldsymbol{s}_{ij} = (s_{u,ij}, s_{v,ij})^T \in \Omega$ is the jth spatial location at time t_i, m_i is the number of spatial locations at t_i, and n is the number of observation times. These spatio-temporal observations are assumed to follow the nonparametric spatio-temporal model

$$y(t, \boldsymbol{s}) = \mu(t, \boldsymbol{s}) + \epsilon(t, \boldsymbol{s}), \quad \text{for } t \in [0, T], \boldsymbol{s} \in \Omega, \tag{1}$$

where $\mu(t, \boldsymbol{s})$ is the mean of $y(t, \boldsymbol{s})$, and $\epsilon(t, \boldsymbol{s})$ is a zero-mean random error. The spatio-temporal data correlation is described by the following covariance function:

$$V(t, t'; \boldsymbol{s}, \boldsymbol{s}') = \text{Cov}\left(y(t, \boldsymbol{s}), y(t', \boldsymbol{s}')\right), \quad \text{for } \ t, t' \in [0, T], \boldsymbol{s}, \boldsymbol{s}' \in \Omega. \tag{2}$$

When $(t, \boldsymbol{s}) = (t', \boldsymbol{s}')$, $V(t, t; \boldsymbol{s}, \boldsymbol{s})$ is just the variance of $y(t, \boldsymbol{s})$, which is denoted as $\sigma^2(t, \boldsymbol{s})$, for any $(t, \boldsymbol{s}) \in [0, T] \times \Omega$. In the above spatio-temporal model (1)-(2), we only assume that the mean and covariance functions $\mu(t,\boldsymbol{s})$ and $V(t,t';\boldsymbol{s},\boldsymbol{s}')$ are continuous functions. Since there are no parametric assumptions imposed on $\mu(t,\boldsymbol{s})$, $V(t,t';\boldsymbol{s},\boldsymbol{s}')$ and the distribution of $\epsilon(t,\boldsymbol{s})$, this model is flexible.

Estimation of the mean function $\mu(t,\boldsymbol{s})$. In this part, we discuss estimation of the mean function $\mu(t,\boldsymbol{s})$ from the observed spatio-temporal data $\{y(t_i,\boldsymbol{s}_{ij}), j=1,\ldots,m_i, i=1,\ldots,n\}$. For any $(t,\boldsymbol{s})\in[0,T]\times\Omega$, Yang and Qiu (2018) suggested estimating $\mu(t,\boldsymbol{s})$ by the following local linear kernel smoothing procedure:

$$\begin{aligned}\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^4}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\left[y(t_i,\boldsymbol{s}_{ij})-\ \theta_\mu-\theta_t(t_i-t)-\theta_u(s_{u,ij}-s_u)-\theta_v(s_{v,ij}-s_v)\right]^2\\ \times K_t\left(\frac{t_i-t}{h_t}\right)K_s\left(\frac{d_E(\boldsymbol{s}_{ij},\boldsymbol{s})}{h_s}\right),\end{aligned}\tag{3}$$

where $\boldsymbol{\theta}=(\theta_\mu,\theta_t,\theta_u,\theta_v)^T$, $\boldsymbol{s}=(s_u,s_v)^T$, h_t and h_s are two bandwidths, $K_t(\cdot)$ and $K_s(\cdot)$ are the Epanechnikov kernel function $K_e(x)=0.75(1-x^2)\mathrm{I}(|x|\leq 1)$ (Epanechnikov 1969), and $d_E(\boldsymbol{s}_{ij},\boldsymbol{s})$ is the Euclidean distance between the two spatial locations $\boldsymbol{s}_{ij}$ and $\boldsymbol{s}$. The solution of (3) to θ_μ is defined to be the estimate of $\mu(t,\boldsymbol{s})$, which has the expression

$$\widehat{\mu}(t,\boldsymbol{s})=\boldsymbol{\zeta}_1^T\left(\mathbf{G}^T\mathbf{W}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{W}\mathbf{Y},\tag{4}$$

where $\boldsymbol{\zeta}_1=(1,0,0,0)^T$, $\mathbf{G}=(\mathbf{G}_{11},\ldots,\mathbf{G}_{nm_n})^T$, $\mathbf{G}_{ij}=(1,(t_i-t),(s_{u,ij}-s_u),(s_{v,ij}-s_v))^T$, $\mathbf{W}=\mathrm{diag}\{w_\mu(1,1;t,\boldsymbol{s}),\ldots,w_\mu(n,m_n;t,\boldsymbol{s})\}$, $w_\mu(i,j;t,\boldsymbol{s})=K_t((t_i-t)/h_t)K_s\left(d_E(\boldsymbol{s}_{ij},\boldsymbol{s})/h_s\right)$ and $\mathbf{Y}=(y(t_1,\boldsymbol{s}_{11}),\,\ldots,\,y(t_n,\boldsymbol{s}_{nm_n}))^T$. From (4), it can be checked that the estimate $\widehat{\mu}(t,\boldsymbol{s})$ is a weighted average of all observations in a local spatio-temporal neighborhood of $(t,\boldsymbol{s})$, where the weights are determined by the two kernel functions and the neighborhood size is controlled by the two bandwidths.

Estimation of the covariance function $V(t,t';\boldsymbol{s},\boldsymbol{s}')$. After the mean function $\mu(t,\boldsymbol{s})$ is estimated by (3) and (4), we can compute the residuals $\widehat{\epsilon}(t_i,\boldsymbol{s}_{ij})=y(t_i,\boldsymbol{s}_{ij})-\widehat{\mu}(t_i,\boldsymbol{s}_{ij})$, for $j=1,\ldots,m_i$ and $i=1,\ldots,n$. Then, the covariance function $V(t,t';\boldsymbol{s},\boldsymbol{s}')$ can be estimated from these residuals by the weighted moment estimation procedure, as discussed in Yang and Qiu (2019). More specifically, the variance function $\sigma^2(t,\boldsymbol{s})=V(t,t;\boldsymbol{s},\boldsymbol{s})$ can be estimated by

$$\widehat{\sigma}^2(t,\boldsymbol{s}) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i}\widehat{\epsilon}^2(t_i,\boldsymbol{s}_{ij})w_\sigma(i,j;t,\boldsymbol{s})}{\sum_{i=1}^{n}\sum_{j=1}^{m_i}w_\sigma(i,j;t,\boldsymbol{s})}, \quad \text{for } (t,\boldsymbol{s})\in[0,T]\times\Omega, \tag{5}$$

where $w_\sigma(i,j;t,\boldsymbol{s}) = K_t\left((t_i-t)/g_t\right)K_s\left(d_E(\boldsymbol{s}_{ij},\boldsymbol{s})/g_s\right)$, for $1\le j\le m_i$ and $1\le i\le n$, and g_t and g_s are two bandwidths that can be different from the bandwidths h_t and h_s. When $(t,\boldsymbol{s})\neq(t',\boldsymbol{s}')$, the covariance function $V(t,t';\boldsymbol{s},\boldsymbol{s}')$ can be estimated by

$$\widehat{V}(t,t';\boldsymbol{s},\boldsymbol{s}') = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=1}^{n}\sum_{l=1}^{m_k}\widehat{\epsilon}(t_i,\boldsymbol{s}_{ij})\widehat{\epsilon}(t_k,\boldsymbol{s}_{kl})w_v(i,j,k,l;t,\boldsymbol{s},t',\boldsymbol{s}')}{\sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=1}^{n}\sum_{l=1}^{m_k}w_v(i,j,k,l;t,\boldsymbol{s},t',\boldsymbol{s}')}, \tag{6}$$

where $w_v(i,j,k,l;t,\boldsymbol{s},t',\boldsymbol{s}') = w_\sigma(i,j;t,\boldsymbol{s})w_\sigma(k,l;t',\boldsymbol{s}')$. Note that the estimates $\widehat{\sigma}^2(t,\boldsymbol{s})$ and $\widehat{V}(t,t';\boldsymbol{s},\boldsymbol{s}')$ obtained by (5) and (6) may not be positive semidefinite, and thus may not be legitimate variance and covariance functions. To address this issue, a modification is needed to make them positive semidefinite. To this end, the matrix-projection-based modification procedure discussed in Yang and Qiu (2019) is used, which can be implemented using the function nearPD() in the R package **Matrix**.

A three-step estimation of $\mu(t,\boldsymbol{s})$ and $V(t,t';\boldsymbol{s},\boldsymbol{s}')$. Although the local linear kernel smoothing procedure in (3)-(4) provides a reliable estimate of $\mu(t,\boldsymbol{s})$, the covariance structure of the observed data is not taken into account in that procedure. Thus, it still has room for improvement. To improve the local linear kernel smoothing estimator defined in (4), Yang and Qiu (2022) developed a three-step local smoothing approach for estimating the spatio-temporal mean and covariance functions. This method consists of three main steps described below. (i) Calculate an initial estimate of $\mu(t,\boldsymbol{s})$ by (3)-(4). (ii) Estimate the spatio-temporal covariance function $V(t,t';\boldsymbol{s},\boldsymbol{s}')$ by (5)-(6). (iii) Calculate the final estimate of $\mu(t,\boldsymbol{s})$ using a weighted local linear kernel smoothing procedure, in which the estimated spatio-temporal covariance function is used. Next, we describe the third step in more details. Let $\widehat{\mathbf{\Sigma}}_{\mathbf{Y}}$ be the estimated covariance matrix of $\mathbf{Y} = (y(t_1,\boldsymbol{s}_{11}), \ldots, y(t_n,\boldsymbol{s}_{nm_n}))^T$ obtained from $\widehat{V}(t,t';\boldsymbol{s},\boldsymbol{s}')$, and $\widehat{\mathbf{\Sigma}}_{\mathbf{K}} = \mathbf{W}^{-1/2}\widehat{\mathbf{\Sigma}}_{\mathbf{Y}}\mathbf{W}^{-1/2}$, where $\mathbf{W} = \text{diag}\{w_\mu(1,1;t,\boldsymbol{s}),\ldots,w_\mu(n,m_n;t,\boldsymbol{s})\}$, $w_\mu(i,j;t,\boldsymbol{s}) = K_t((t_i-t)/b_t)K_s\left(d_E(\boldsymbol{s}_{ij},\boldsymbol{s})/b_s\right)$, and b_t and b_s are two bandwidths that can be different from the bandwidths h_t and h_s used in the first step for computing the initial estimate of $\mu(t,\boldsymbol{s})$. Then, the final estimate of $\mu(t,\boldsymbol{s})$

is defined to be

$$\widetilde{\mu}(t, \boldsymbol{s}) = \boldsymbol{\zeta}_1^T \left(\mathbf{G}^T \widehat{\boldsymbol{\Sigma}}_{\mathbf{K}}^{-1} \mathbf{G} \right)^{-1} \mathbf{G}^T \widehat{\boldsymbol{\Sigma}}_{\mathbf{K}}^{-1} \mathbf{Y}, \tag{7}$$

where $\boldsymbol{\zeta}_1$ and $\mathbf{G}$ are defined immediately after (4). From (7), it can be seen that the final estimate $\widetilde{\mu}(t, \boldsymbol{s})$ has taken into account spatio-temporal data correlation through $\widehat{\boldsymbol{\Sigma}}_{\mathbf{Y}}$ in $\widehat{\boldsymbol{\Sigma}}_{\mathbf{K}}$.

2.2 Online monitoring of spatio-temporal data

In this part, we provide a detailed description of the online spatio-temporal process monitoring methods suggested in Yang and Qiu (2020) and Qiu and Yang (2021 and 2023). Assume that an in-control (IC) dataset $\{y(t_i, \boldsymbol{s}_{ij}), j = 1, \ldots, m_i, i = 1, \ldots, n\}$ has been collected before online process monitoring. After the IC spatio-temporal model defined in (1)-(2) is estimated from the IC data, the regular (or IC) spatio-temporal pattern of the process under monitoring can be described by the estimated mean, variance, and covariance functions $\widehat{\mu}(t, \boldsymbol{s})$, $\widehat{\sigma}^2(t, \boldsymbol{s})$ and $\widehat{V}(t, t; \boldsymbol{s}, \boldsymbol{s}')$. In practice, it is often reasonable to assume that the IC spatio-temporal pattern is periodic in time due to seasonality, and $[0, T]$ covers at least one entire period. In such applications, $\widehat{\mu}(t, \boldsymbol{s})$, $\widehat{\sigma}^2(t, \boldsymbol{s})$ and $\widehat{V}(t, t; \boldsymbol{s}, \boldsymbol{s}')$ can be extended in the time domain from $[0, T]$ to $[0, \infty)$ to describe the regular spatio-temporal pattern of the process. Our major goal of online process monitoring is to detect any shifts in the observed spatio-temporal pattern of the process from the regular spatio-temporal pattern.

Data decorrelation and standardization. Assume that the spatio-temporal observations to monitor at time $t_i^* \in (T, \infty)$, for $i \geq 1$, are collected at spatial locations $\{s_{ij}^* \in \Omega, j = 1, \ldots, m_i^*\}$. These observations are denoted as $\{y(t_i^*, \boldsymbol{s}_{ij}^*), j = 1, \ldots, m_i^*, i \geq 1\}$. Like observations in the IC data, the observations $\{y(t_i^*, \boldsymbol{s}_{ij}^*), j = 1, \ldots, m_i^*, i \geq 1\}$ could be spatio-temporally correlated, and their IC means could vary across both space and time. Thus, they cannot be monitored effectively by conventional statistical process control (SPC) charts, because the conventional charts are designed for cases with independent and identically distributed IC process observations. In the SPC literature, it has been well discussed that the conventional charts are unreliable and can give misleading results when they are used for monitoring serially correlated data (cf., Apley and Tsung 2002, Qiu et al. 2020). To overcome this limitation, we can sequentially decorrelate the observed data before online process monitoring, using the following data decorrelation and standardization procedure:

- When $i = 1$, let $\mathbf{Y}(t_1^*) = (y(t_1^*, \boldsymbol{s}_{11}^*), \ldots, y(t_1^*, \boldsymbol{s}_{1m_1^*}^*))^T$, $\widehat{\epsilon}(t_1^*, \boldsymbol{s}_{1j}^*) = y(t_1^*, \boldsymbol{s}_{1j}^*) - \widehat{\mu}(t_1^*, \boldsymbol{s}_{1j}^*)$, for $1 \leq j \leq m_1^*$, and $\widehat{\boldsymbol{\epsilon}}(t_1^*) = (\widehat{\epsilon}(t_1^*, \boldsymbol{s}_{11}^*), \ldots, \widehat{\epsilon}(t_1^*, \boldsymbol{s}_{1m_1^*}^*))^T$. Then, the decorrelated data at time t_1^* is defined to be $\widehat{\boldsymbol{e}}(t_1^*) = (\widehat{e}(t_1^*, \boldsymbol{s}_{11}^*), \ldots, \widehat{e}(t_1^*, \boldsymbol{s}_{1m_1^*}^*))^T = \widehat{\boldsymbol{\Sigma}}_1^{-1/2} \widehat{\boldsymbol{\epsilon}}(t_1^*)$, where $\widehat{\boldsymbol{\Sigma}}_1$ is an estimate of $\boldsymbol{\Sigma}_1 = \text{Cov}\left(\mathbf{Y}(t_1^*), \mathbf{Y}(t_1^*)\right)$ obtained from $\widehat{V}(t, t'; \boldsymbol{s}, \boldsymbol{s}')$.

- When $i > 1$, let $\mathbf{Y}(t_i^*) = (y(t_i^*, \boldsymbol{s}_{i1}^*), \ldots, y(t_i^*, \boldsymbol{s}_{im_i^*}^*))^T$, $\mathbf{Y}_{i-1} = \left(\mathbf{Y}^T(t_1^*), \ldots, \mathbf{Y}^T(t_{i-1}^*)\right)^T$, $\widehat{\epsilon}(t_i^*, \boldsymbol{s}_{ij}^*) = y(t_i^*, \boldsymbol{s}_{ij}^*) - \widehat{\mu}(t_i^*, \boldsymbol{s}_{ij}^*)$, for $1 \leq j \leq m_i^*$, $\widehat{\boldsymbol{\epsilon}}(t_i^*) = (\widehat{\epsilon}(t_i^*, \boldsymbol{s}_{i1}^*), \ldots, \widehat{\epsilon}(t_i^*, \boldsymbol{s}_{im_i^*}^*))^T$, and $\widehat{\boldsymbol{\epsilon}}_{i-1} = (\widehat{\boldsymbol{\epsilon}}^T(t_1^*), \ldots, \widehat{\boldsymbol{\epsilon}}^T(t_{i-1}^*))^T$. Then, the decorrelated data at time t_i^* is

  $$\widehat{\boldsymbol{e}}(t_i^*) = \left(\widehat{e}(t_i^*, \boldsymbol{s}_{i1}^*), \ldots, \widehat{e}(t_i^*, \boldsymbol{s}_{im_i^*}^*)\right)^T = \widehat{\boldsymbol{\Sigma}}_{ii \cdot i-1}^{-1/2} \left(\widehat{\boldsymbol{\epsilon}}(t_i^*) - \widehat{\boldsymbol{\Sigma}}_{i-1,i}^T \widehat{\boldsymbol{\Sigma}}_{i-1,i-1}^{-1} \widehat{\boldsymbol{\epsilon}}_{i-1}\right),$$

  where $\widehat{\boldsymbol{\Sigma}}_{ii \cdot i-1} = \widehat{\boldsymbol{\Sigma}}_i - \widehat{\boldsymbol{\Sigma}}_{i-1,i}^T \widehat{\boldsymbol{\Sigma}}_{i-1,i-1}^{-1} \widehat{\boldsymbol{\Sigma}}_{i-1,i}$, and $\widehat{\boldsymbol{\Sigma}}_i$, $\widehat{\boldsymbol{\Sigma}}_{i-1,i}$ and $\widehat{\boldsymbol{\Sigma}}_{i-1,i-1}$ are the estimates of $\boldsymbol{\Sigma}_i = \text{Cov}\left(\mathbf{Y}(t_i^*), \mathbf{Y}(t_i^*)\right)$, $\boldsymbol{\Sigma}_{i-1,i} = \text{Cov}\left(\mathbf{Y}_{i-1}, \mathbf{Y}(t_i^*)\right)$ and $\boldsymbol{\Sigma}_{i-1,i-1} = \text{Cov}\left(\mathbf{Y}_{i-1}, \mathbf{Y}_{i-1}\right)$, respectively, all of which are obtained from $\widehat{V}(t, t'; \boldsymbol{s}, \boldsymbol{s}')$.

After using the above data decorrelation and standardization algorithm, the original spatio-temporal observations $\{y(t_i^*, \boldsymbol{s}_{ij}^*), j = 1, \ldots, m_i^*, i \geq 1\}$ are transformed to the decorrelated and standardized observations $\{\widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*), j = 1, \ldots, m_i^*, i \geq 1\}$. If the process under monitoring is IC, then these decorrelated observations should be asymptotically uncorrelated with each other, and the asymptotic mean and variance of each of them should be 0 and 1, respectively. Then, process monitoring can focus on these decorrelated and standardized observations rather than the original observations.

Online monitoring of a spatial process. From the description in the previous part, it can be seen that each decorrelated and standardized observation $\widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*)$ is a linear combination of the original process observations. Thus, its IC distribution should be close to the standard normal distribution under some regularity conditions by the central limit theorem. Consequently, the asymptotic distribution of $\sum_{j=1}^{m_i^*} \widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*)^2$ should be the χ^2 distribution with the degrees of freedom of m_i^*, and the asymptotic mean and variance of $\left(\sum_{j=1}^{m_i^*} \widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*)^2 - m_i^*\right)/\sqrt{2m_i^*}$ should be 0 and 1, respectively. Then, Yang and Qiu (2020) suggested the following cumulative sum (CUSUM) chart for online process monitoring:

$$C_i = \max\left\{0, C_{i-1} + \frac{\sum_{j=1}^{m_i^*} \widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*)^2 - m_i^*}{\sqrt{2m_i^*}} - \gamma\right\}, \quad \text{for } i \geq 1, \tag{8}$$

and the chart gives a signal at time t_i^* if the charting statistic C_i exceeds a control limit L, where $C_0 = 0$ and $\gamma > 0$ is often called a reference value or an allowance constant (cf., Qiu 2014, Chapter 4).

Usually, the performance of a control chart is evaluated by the IC average run length (ARL), denoted as ARL_0 and defined as the average number of observation time points from the beginning of process monitoring to a signal by the chart when the process is IC, and the out-of-control (OC) ARL, denoted as ARL_1 and defined as the average number of observation time points from the occurrence of a process distributional shift to a signal time by the chart. When designing a control chart like the one in (8), the ARL_0 value is usually pre-specified at a given level and its control limit is chosen to achieve the pre-specified ARL_0 value. Then, the chart performs better if its ARL_1 value is smaller for detecting a given shift. See a systematic discussion about the design of a control chart in Qiu (2014).

To use the CUSUM chart (8), we need to specify the allowance constant γ. In the literature, it has been well studied that a large γ value is good for detecting large shifts and a small γ value is good for detecting small shifts (Qiu 2014, Chapter 4). Commonly used γ values include 0.1, 0.2, 0.3, 0.5 and 1. For a pre-specified ARL_0 value, we also need to determine the control limit L such that the pre-specified ARL_0 value is reached. To this end, the block bootstrap procedure described in Section B of the supplementary file can be used.

Online monitoring of a spatial process using covariate information. The CUSUM chart (8) provides a reliable tool for online monitoring of a spatial process. In many applications, the spatio-temporal response $y(t, \boldsymbol{s})$ is associated with some covariates. For example, disease incidence can be affected by many risk factors including air temperature, humidity, and other weather and/or environmental conditions. In such applications, it is our belief that proper use of the covariate information can improve the performance of the related control chart. Next, we introduce the exponentially weighted moving average (EWMA) chart suggested by Qiu and Yang (2021) that can make use of covariate information. This chart is denoted as EWMAC hereafter, where the last letter "C" denotes "covariates". The EWMAC method consists of two main steps. First, a semiparametric spatio-temporal model is fitted from an IC dataset to estimate the regular spatio-temporal pattern of the process under monitoring in presence of the related covariates. Second, the EWMAC chart is developed for online process monitoring, which can accommodate the covariate information. Next, we provide more details about these two steps.

Let $\{y(t_i, \boldsymbol{s}_{ij}), j = 1, \ldots, m_i, i = 1, \ldots, n\}$ be an IC dataset obtained before online process monitoring. Observations in this dataset are assumed to follow the semiparametric spatio-temporal model

$$y(t, \boldsymbol{s}) = \mu(t, \boldsymbol{s}) + \mathbf{X}_1^T(t)\boldsymbol{\beta}_1 + \mathbf{X}_2^T(t, \boldsymbol{s})\boldsymbol{\beta}_2 + \epsilon(t, \boldsymbol{s}), \quad \text{for } (t, \boldsymbol{s}) \in [0, T] \times \Omega, \tag{9}$$

where $\mathbf{X}_1(t)$ is a vector of p_1 time-dependent covariates, $\mathbf{X}_2(t, \boldsymbol{s})$ is a vector of p_2 space/time-dependent covariates, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are their regression coefficients, $\mu(t, \boldsymbol{s})$ is the mean of $y(t, \boldsymbol{s})$ after excluding the part explained by $\mathbf{X}_1(t)$ and $\mathbf{X}_2(t, \boldsymbol{s})$, and $\epsilon(t, \boldsymbol{s})$ is a zero-mean random error, for any $(t, \boldsymbol{s}) \in [0, T] \times \Omega$. It should be pointed out that there could be time-independent covariates (they could depend on space) that are associated with $y(t, \boldsymbol{s})$ in practice. Such covariates are not included explicitly in the model (9) because they would not provide any helpful information about the temporal variation of $y(t, \boldsymbol{s})$ and thus are not helpful for online process monitoring.

To estimate the regression coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T)^T$, Qiu and Yang (2021) proposed the following iterative algorithm:

(i) Set $\boldsymbol{\beta} = \mathbf{0}$ and obtain an initial estimate of $\mu(t, \boldsymbol{s})$ by (4), denoted as $\widehat{\mu}^{(0)}(t, \boldsymbol{s})$.

(ii) In the kth iteration, for $k \geq 1$, implement the following two steps:

(a) Compute the least squares estimate of $\boldsymbol{\beta}$, denoted as $\widehat{\boldsymbol{\beta}}^{(k)}$, from the linear model $Z^{(k)}(t,\boldsymbol{s}) = \mathbf{X}_1^T(t)\boldsymbol{\beta}_1 + \mathbf{X}_2^T(t,\boldsymbol{s})\boldsymbol{\beta}_2 + \epsilon(t,\boldsymbol{s})$, where $Z^{(k)}(t,\boldsymbol{s}) = y(t,\boldsymbol{s}) - \widehat{\mu}^{(k-1)}(t,\boldsymbol{s})$.

(b) Update the estimate of $\mu(t,\boldsymbol{s})$ by replacing $\mathbf{Y}$ in (4) by $\mathbf{Y}^{(k)} = (y^{(k)}(t_1,\boldsymbol{s}_{11}),\ldots, y^{(k)}(t_n,\boldsymbol{s}_{nm_n}))^T$, where $y^{(k)}(t_i,\boldsymbol{s}_{ij}) = y(t_i,\boldsymbol{s}_{ij}) - \mathbf{X}_1^T(t_i)\widehat{\boldsymbol{\beta}}_1^{(k)} - \mathbf{X}_2^T(t_i,\boldsymbol{s}_{ij})\widehat{\boldsymbol{\beta}}_2^{(k)}$, for each i and j. The updated estimate of $\mu(t,\boldsymbol{s})$ is denoted as $\widehat{\mu}^{(k)}(t,\boldsymbol{s})$.

(iii) The iterative algorithm stops when $\|\widehat{\boldsymbol{\beta}}^{(k)} - \widehat{\boldsymbol{\beta}}^{(k-1)}\|_1/\|\widehat{\boldsymbol{\beta}}^{(k-1)}\|_1 \leq \varsigma$, where $\varsigma > 0$ is a pre-specified small number and $\|\boldsymbol{x}\|_1$ denotes the summation of absolute values of all elements in the vector $\boldsymbol{x}$. Then, $\widehat{\boldsymbol{\beta}}^{(k)}$ and $\widehat{\mu}^{(k)}(t,\boldsymbol{s})$ are the final estimates of $\boldsymbol{\beta}$ and $\mu(t,\boldsymbol{s})$, respectively. These final estimates are also denoted as $\widehat{\boldsymbol{\beta}}$ and $\widehat{\mu}(t,\boldsymbol{s})$.

By using the above iterative algorithm, we obtain the estimated regression coefficients $\widehat{\boldsymbol{\beta}}_1$ and $\widehat{\boldsymbol{\beta}}_2$. Let $z(t,\boldsymbol{s}) = \mathbf{X}_1^T(t)\widehat{\boldsymbol{\beta}}_1 + \mathbf{X}_2^T(t,\boldsymbol{s})\widehat{\boldsymbol{\beta}}_2$ denote the estimated covariate effect on the response $y(t,\boldsymbol{s})$, $\mu_y(t,\boldsymbol{s})$ and $V_y(t,t';\boldsymbol{s},\boldsymbol{s}')$ be the mean and covariance functions of $y(t,\boldsymbol{s})$, and $\mu_z(t,\boldsymbol{s})$ and $V_z(t,t';\boldsymbol{s},\boldsymbol{s}')$ be the mean and covariance functions of $z(t,\boldsymbol{s})$. These functions can be estimated from the IC dataset by using (3)-(4) and (5)-(6), and their respective estimates are denoted as $\widehat{\mu}_y(t,\boldsymbol{s})$, $\widehat{V}_y(t,t';\boldsymbol{s},\boldsymbol{s}')$, $\widehat{\mu}_z(t,\boldsymbol{s})$ and $\widehat{V}_z(t,t';\boldsymbol{s},\boldsymbol{s}')$.

Next, we provide a detailed description of the EWMAC chart for detecting an upward mean shift in the spatio-temporal process under monitoring. The EWMAC chart for detecting downward shifts can be discussed similarly. Assume that the spatio-temporal observations to monitor are observed at times t_i^* and locations $\{\boldsymbol{s}_{ij}^*, j=1,\ldots,m_i^*\}$, for $i \geq 1$. These observations are denoted as $\{y(t_i^*,\boldsymbol{s}_{ij}^*), j=1,\ldots,m_i^*, i\geq 1\}$, and the related observations of the time-dependent and space/time-dependent covariates are denoted as $\{\mathbf{X}_1(t_i^*), i\geq 1\}$ and $\{\mathbf{X}_2(t_i^*,\boldsymbol{s}_{ij}^*), j=1,\ldots,m_i^*, i\geq 1\}$, respectively. Before monitoring the spatio-temporal process, let us first apply the data decorrelation and standardization algorithm discussed before to both $\{y(t_i^*,\boldsymbol{s}_{ij}^*)\}$ and $\{z(t_i^*,\boldsymbol{s}_{ij}^*)\}$, where $z(t_i^*,\boldsymbol{s}_{ij}^*) = \mathbf{X}_1^T(t_i^*)\widehat{\boldsymbol{\beta}}_1 + \mathbf{X}_2^T(t_i^*,\boldsymbol{s}_{ij}^*)\widehat{\boldsymbol{\beta}}_2$, and their decorrelated data are denoted as $\{\widehat{e}_y(t_i^*,\boldsymbol{s}_{ij}^*)\}$ and $\{\widehat{e}_z(t_i^*,\boldsymbol{s}_{ij}^*)\}$. Then, the following

EWMA chart is used for detecting upward shifts in the process $z(t, \boldsymbol{s})$:

$$E_{z,i} = \lambda \frac{\sum_{j=1}^{m_i^*} \widehat{e}_z(t_i^*, \boldsymbol{s}_{ij}^*)}{\sqrt{m_i^*}} + (1-\lambda)E_{z,i-1}, \quad \text{for } i \geq 1,$$

where $\lambda \in (0, 1]$ is a weighting parameter and $E_{z,0} = 0$. If there is an upward mean shift in $z(t, \boldsymbol{s})$ at or before the time t_i^*, then the value of $E_{z,i}$ would be relatively large because of the shift. Therefore, the EWMA charting statistic $E_{z,i}$ provides a measure of the likelihood of an upward mean shift in $z(t, \boldsymbol{s})$. In the spatio-temporal process monitoring problem, our ultimate goal is to detect shifts in $y(t, \boldsymbol{s})$, which may or may not be caused by shifts in $z(t, \boldsymbol{s})$. In addition, shifts in $z(t, \boldsymbol{s})$ are not our major concern in the current process monitoring problem, although any helpful information in $z(t, \boldsymbol{s})$ should be used in process monitoring. By these considerations, Qiu and Yang (2021) suggested the following EWMAC chart for online monitoring of $y(t, \boldsymbol{s})$:

$$E_{y,i} = \mathrm{W}(E_{z,i}; \lambda, \kappa) \frac{\sum_{j=1}^{m_i^*} \widehat{e}_y(t_i^*, \boldsymbol{s}_{ij}^*)}{\sqrt{m_i^*}} + \left[1 - \mathrm{W}(E_{z,i}; \lambda, \kappa)\right] E_{y,i-1}, \quad \text{for } i \geq 1, \tag{10}$$

where $E_{y,0} = 0$, and $\mathrm{W}(E_{z,i}; \lambda, \kappa) \in (0, 1]$ is a weighting parameter that depends on the covariate charting statistic $E_{z,i}$ and two parameters $\lambda \in (0, 1]$ and $\kappa > 0$. The chart gives a signal at time t_i^* if $E_{y,i} > L$, where $L > 0$ is a control limit.

Online monitoring of a spatial process using exponentially weighted spatial LASSO. In many spatio-temporal process monitoring applications, a systematic process shift starts in some small regions that are spatially clustered (Wang et al. 2018). This kind of spatial feature has not been taken into account in the CUSUM chart (8) and the EWMAC chart (10) because they treat the spatial locations equally in their construction. To address this issue, Qiu and Yang (2023) proposed an effective method for spatio-temporal process monitoring by combining the ideas of regression modeling in the space domain by spatial LASSO and exponentially weighted data smoothing in the time domain. The related control chart is denoted as EWSL, representing "exponentially weighted spatial LASSO". Some details about this chart are given below.

For online process monitoring, a basic strategy is to use as much available information about the process under monitoring as possible and give less weights to process observations that are collected farther away from the current observation time in the decision-making process. On the other hand, the spatial feature that process shifts usually start in small clustered spatial regions should be taken into account as well. Based on these considerations, Qiu and Yang (2023) proposed the EWSL chart that combined the following two procedures. First, a spatial LASSO procedure is proposed to find small clustered regions with possible shifts in the process distribution at the current observation time, which can accommodate complex spatial data structures. Second, an exponentially weighted smoothing procedure is suggested in the temporal domain to make use of all historical spatio-temporal data. These two procedures are then combined seamlessly in the spatio-temporal domain for online process monitoring. More specifically, by combining the spatial LASSO idea with the exponentially weighted data smoothing idea, Qiu and Yang (2023) proposed the following penalized exponentially weighted kernel smoothing procedure for estimating the means of $\{\widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*), j = 1, \ldots, m_i^*\}$ at time t_i^*:

$$\begin{aligned}&\underset{\theta_1,\ldots,\theta_{m_i^*}}{\arg\min} \sum_{j=1}^{m_i^*}\sum_{k=1}^{i}\sum_{l=1}^{m_k^*} \left[\widehat{e}(t_k^*, \boldsymbol{s}_{kl}^*) - \theta_j\right]^2 K_s\left(d_E(\boldsymbol{s}_{kl}^*, \boldsymbol{s}_{ij}^*)/h\right)(1-\lambda)^{(t_i^*-t_k^*)}\\ &+ \psi_1 \sum_{j=1}^{m_i^*} \varpi_{1j}|\theta_j| + \psi_2\sum_{j=1}^{m_i^*}\varpi_{2j}\left|\theta_j - \frac{\sum_{1\le l\le m_i^*} K_s(d_E(\boldsymbol{s}_{ij}^*, \boldsymbol{s}_{il}^*)/h)\theta_l}{\sum_{1\le l\le m_i^*} K_s(d_E(\boldsymbol{s}_{ij}^*, \boldsymbol{s}_{il}^*)/h)}\right|,\end{aligned} \tag{11}$$

where $h > 0$ is a bandwidth, $K_s(\cdot)$ is the Epanechnikov kernel function, $\psi_1, \psi_2 > 0$ are two tuning parameters, $\varpi_{1j} = \left|\widetilde{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{ij}^*)\right|^{-1}$ and

$$\varpi_{2j} = \left|\widetilde{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{ij}^*) - \frac{\sum_{1\le l\le m_i^*} K_s(d_E(\boldsymbol{s}_{ij}^*, \boldsymbol{s}_{il}^*)/h)\widetilde{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{il}^*)}{\sum_{1\le l\le m_i^*} K_s(d_E(\boldsymbol{s}_{ij}^*, \boldsymbol{s}_{il}^*)/h)}\right|^{-1}$$

are the adaptive LASSO weights (cf., Zou 2006), and $\widetilde{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{ij}^*)$ is the estimated mean of $\widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*)$ obtained by (11) after choosing $\psi_1 = \psi_2 = 0$. Then, the solution of (11) to θ_j, denoted as $\widehat{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{ij}^*)$, is defined to be the estimated mean of $\widehat{e}(t_i^*, \boldsymbol{s}_{ij}^*)$, for $j = 1, \ldots, m_i^*$.

It can be seen from (11) that the estimates $\{\widehat{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{ij}^*), j = 1, \ldots, m_i^*\}$ have taken into account the information from previous process observations. Due to the shrinkage property of the spatial LASSO penalty in (11) (i.e., the last two terms in that expression), many elements in $\{\widehat{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{ij}^*)\}$ would be zero and only those at locations with process shifts would be nonzero. So, to detect process shifts, it is natural to consider the statistic $Q_i = \widehat{\boldsymbol{\mu}}_{\widehat{e},i}^T \widehat{\Sigma}_{\widehat{\boldsymbol{\mu}}_{\widehat{e},i}}^{-1} \widehat{\boldsymbol{\mu}}_{\widehat{e},i}$, where $\widehat{\boldsymbol{\mu}}_{\widehat{e},i} = (\widehat{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{i1}^*), \ldots, \widehat{\mu}_{\widehat{e}}(t_i^*, \boldsymbol{s}_{im_i^*}^*))^T$, and $\widehat{\Sigma}_{\widehat{\boldsymbol{\mu}}_{\widehat{e},i}}$ is an estimate of the covariance matrix of $\widehat{\boldsymbol{\mu}}_{\widehat{e},i}$. Then, the charting statistic of EWSL is the following standardized version of Q_i:

$$SQ_i = \frac{Q_i - \widehat{\mathrm{E}}(Q_i)}{\sqrt{\widehat{\mathrm{Var}}(Q_i)}}, \quad \text{for } i \geq 1, \tag{12}$$

where $\widehat{\mathrm{E}}(Q_i)$ and $\widehat{\mathrm{Var}}(Q_i)$ are the estimates of the mean and variance of Q_i, respectively. The chart (12) gives a signal at time t_i^* if $SQ_i > L$, where $L > 0$ is a control limit.

3 Description of the R Package SpTe2M

The R Package **SpTe2M** has been developed recently to implement spatio-temporal data modeling and monitoring methods described in Section 2. The package can be downloaded directly from the Comprehensive R Archive Network (CRAN) with the web address https://cran.r-project.org/. After the package is downloaded to your computer, it can be installed using the following R commands.

```
R> install.packages("SpTe2M")
R> library("SpTe2M")
```

In the current version of the **SpTe2M** package, there are seven main functions that are summarized in Table 1. The package manual contains additional information for all these functions. This documentation can be accessed conveniently in R using the following commands:

```
R> help(package = "SpTe2M") # Access the package manual
R> ?spte_meanest # Read documentation about the spte_meanest() function
```

The package also includes a vignette to illustrate how to implement the main functions in a real data application. Users can run the R code below to have access to the vignette.

```
R> vignette('SpTe2M',package='SpTe2M') # Read the vignette
```

Table 1: Main functions in the **SpTe2M** package.

Function	Description
spte_meanest()	Estimate the spatio-temporal mean function
spte_covest()	Estimate the spatio-temporal covariance function
spte_decor()	Decorrelate the spatio-temporal data
sptemnt_cusum()	Spatio-temporal process monitoring by the CUSUM chart (8)
spte_semiparmreg()	Fit the semiparametric spatio-temporal model (9)
sptemnt_ewmac()	Spatio-temporal process monitoring using the EWMAC chart (10)
sptemnt_ewsl()	Spatio-temporal process monitoring using the EWSL chart (12)

Demonstration of the package SpTe2M using the Florida ILI dataset. In the next two subsections, the functionality of the package **SpTe2M** is demonstrated using the Florida influenza-like illness (ILI) data that were collected by the Electronic Surveillance System for the Early Notification of Community-based Epidemics (ESSENCE) of the Florida Department of Health (FDOH). ILI is a respiratory infection caused by a variety of influenza viruses. Patients with ILI usually have severe respiratory illness with fever, cough, sore throat, and even difficulty in breathing. In the US, it is reported that 15-40% of the population develop illness from influenza every year. Among those ILI patients, there are about 36,000 hospitalizations and 114,000 deaths due to influenza infection (Fiore et al. 2010). Previously, the method to estimate the incidence rate of ILI is to carry out repeated seroprevalence surveys that are resource-intensive and slow. Thus, it is unfeasible for early detection of ILI outbreaks. To overcome this difficulty, FDOH built a syndromic surveillance system (i.e., ESSENCE) for collecting near real-time pre-diagnostic data from 264 emergency departments and urgent care centers that are distributed in all counties of Florida. As a demonstration, the ILI incidence rates collected by the ESSENCE system for all 67 Florida counties on 06/15 (a summer time) and 12/15 (a winter time) in the three years 2012-2014 are presented in Figure 1. It can be seen from the figure that ILI incidence rates have seasonal patterns with more ILI cases in the winters and fewer ILI cases in the summers. Moreover, there seems to be an unusual pattern of ILI incidence rates in the winter of 2014 because the incidence rates on 12/15/2014 are much higher than those on 12/15/2012 and 12/15/2013. Next, we will demonstrate the use of different functions in the **SpTe2M** package using this built-in ILI dataset that contains the observed daily ILI incidence rates at 67 Florida counties during 2012-2014. In the literature, spatial data can roughly classified into the following three categories: areal data, geospatial data, and point process data (cf., Cressie and Wikle 2011). The ILI dataset belongs to the category of areal data since observations included in the dataset are measured for 67 geographical regions. In addition to observations of the variable Rate (i.e., ILI incidence rate), the dataset also contains observations of the following 7 variables: County, Date, Lat (i.e., latitude of the centroid of a county), Long (i.e., longitude of the centroid of a county), Time, Temp (i.e., air temperature), and RH (i.e., relative humidity).

3.1 Spatio-temporal data modeling

This subsection demonstrates how to estimate the mean and covariance functions based on the local linear kernel smoothing procedure (3)-(4) and the weighted moment estimation procedure (5)-(6) by using the functions spte_meanest() and spte_covest() in **SpTe2M**. The arguments of the function spte_meanest() are listed below:

- y: A vector of spatio-temporal observations of the response y.
- st: A 3-column matrix specifying the locations and times of the observations in y.
- ht: Optional user-supplied bandwidth h_t used in (3). Its default value is NULL. If ht=NULL, then it will be chosen by the modified cross-validation procedure defined in (A1)-(A2) of the supplementary file.

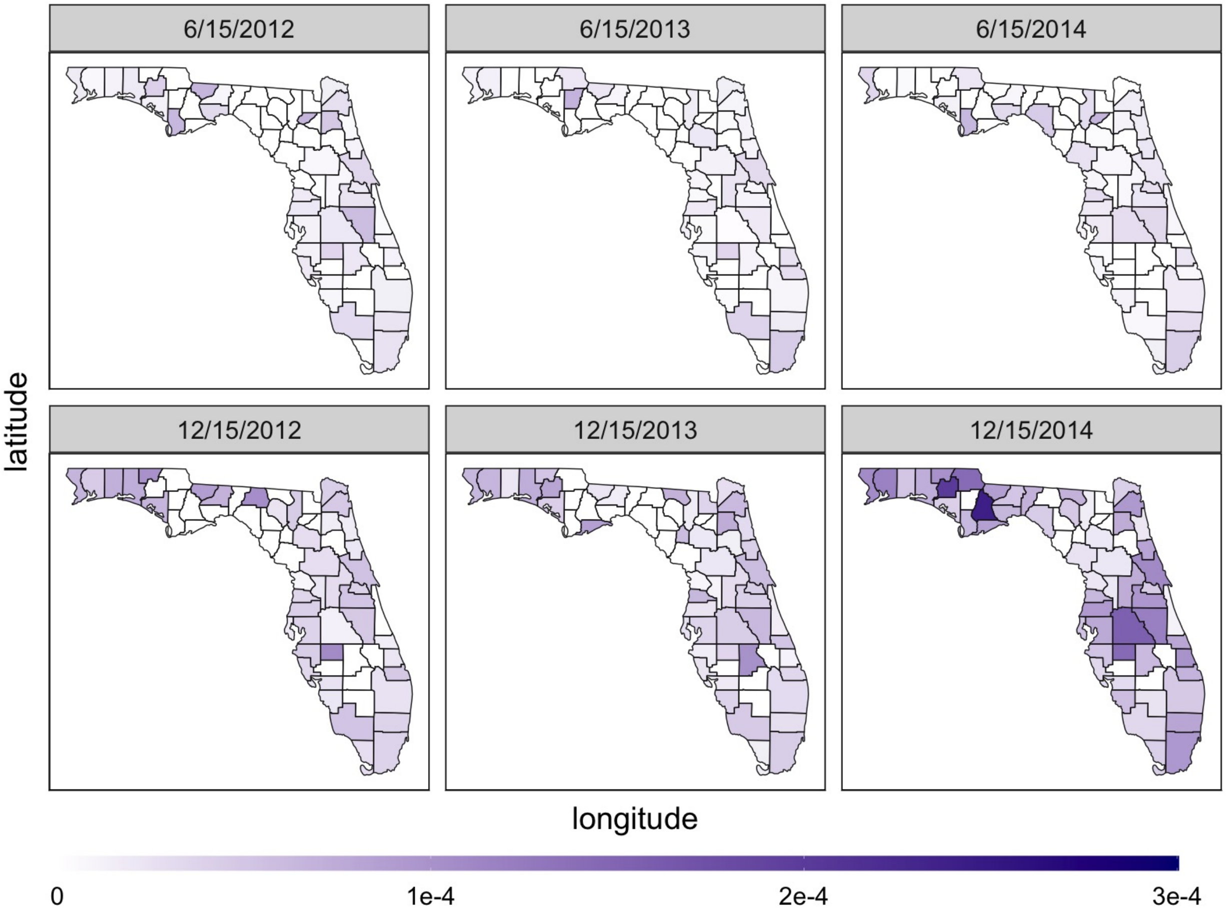

Figure 1: Observed ILI incidence rates in Florida on 06/15 and 12/15 in the three years 2012-2014.

- hs: Optional user-supplied bandwidth h_s used in (3). Its default value is NULL. If hs=NULL, then it will be chosen by the modified cross-validation procedure.

- cor: It indicates whether we should accommodate the covariance structure when estimating the mean function. If cor=FALSE, then the local linear kernel smoothing procedure (3)-(4) is used. If cor=TRUE, then the three-step local smoothing estimate (7) is used. Its default is FALSE.

- stE: A three-column matrix specifying the spatial locations and times where we want to calculate the estimated values of the mean function. Its default is NULL. In such a case, stE=st.

The output of the function spte_meanest() is a list that contains three elements bandwidth, stE and muhat, which correspond to the bandwidths (ht, hs), the times and spatial locations, and the related mean estimates, respectively.

As an example, the command below estimates the mean function of the ILI data in the year 2013 by the local linear kernel smoothing procedure (3)-(4).

```
R> data(ili_dat) # Load the ILI data
R> n <- 365; m <- 67 # Specify the number of observation times (i.e., 365)
                     # and the number of observation locations (i.e., 67
                     # counties) at each time.
R> N1 <- (1+n)*m; N2 <- n*m # Specify the numbers of observations
                            # in years 2012 and 2013
R> subdat <- ili_dat[(N1+1):(N1+N2),] # Obtain the ILI data in 2013
R> y.sub <- subdat$Rate; st.sub <- subdat[,c('Lat','Long','Time')]
R> mu.est <- spte_meanest(y=y.sub,st=st.sub)
```

In the above example, we do not specify the arguments ht, hs and cor when using the function spte_meanest(). So, by default, the local linear kernel smoothing procedure (3)-(4) is used for estimating the spatio-temporal mean function, and the bandwidths (h_t, h_s) involved in the estimation procedure are chosen by the modified cross-validation procedure in (A1)-(A2) of the supplementary file. Of course, we can also specify values of (h_t, h_s) by using the arguments ht and hs. In addition, if we prefer to use the three-step local smoothing procedure (7) to estimate the mean function, then we can set cor=TRUE. In such cases, the bandwidths (b_t, b_s) involved in the three-step estimation procedure would be chosen by minimizing the value of $\text{EMSE}(b_t, b_s)$ defined in (A5) of the supplementary file.

The estimated means can be extracted by using the command mu.est$muhat. To visually check whether they describe the observed ILI data well, we use the code below to plot the estimated means for the Broward county in the top-left panel of Figure 2, along with the observed ILI incidence rates of that county.

```
R> mu <- mu.est$muhat; mu <- t(matrix(mu,m,n)); obs <- t(matrix(y.sub,m,n))
R> id <- 6 # the 6th county in the ILI dataset is the Broward county
R> plot(1:365,mu[,id],type='l',lty=1,lwd=1.5,ylim=c(0,8e-5))
R> points(1:365,obs[,id])
```

The estimated mean values for three other counties Lake, Pinellas and Seminole are also presented in Figure 2. From the figure, it can be seen that the estimated spatio-temporal mean function can well describe the longitudinal pattern of the observed ILI incidence rates. Thus, the function spte_meanest() can indeed provide a reliable tool for estimating the spatio-temporal mean function.

Next, we describe how to estimate the spatio-temporal covariance $V(t,t';\boldsymbol{s},\boldsymbol{s}')$ by using the function spte_covest(). A typical spte_covest() code snippet looks as follows:

```
spte_covest(y,st,gt=NULL,gs=NULL,stE1=NULL,stE2=NULL)
```

Regarding the above command, the auguments y and st in spte_covest() are the same as those in spte_meanest() explained earlier. The rest of the arguments are described in Table 2. The output of spte_covest() is also a list, and its element covhat is the estimated covariance. To estimate the covariance of the ILI data in the year 2013 by the weighted moment estimation procedure (5)-(6), we can use the following code:

```
R> cov.est <- spte_covest(y=y.sub,st=st.sub)
```

In the above command, we did not specify the arguments gt and gs. In such cases, the two bandwidths (g_t, g_s) would be determined by minimizing the mean squared prediction error defined in (A3) of the supplementary file.

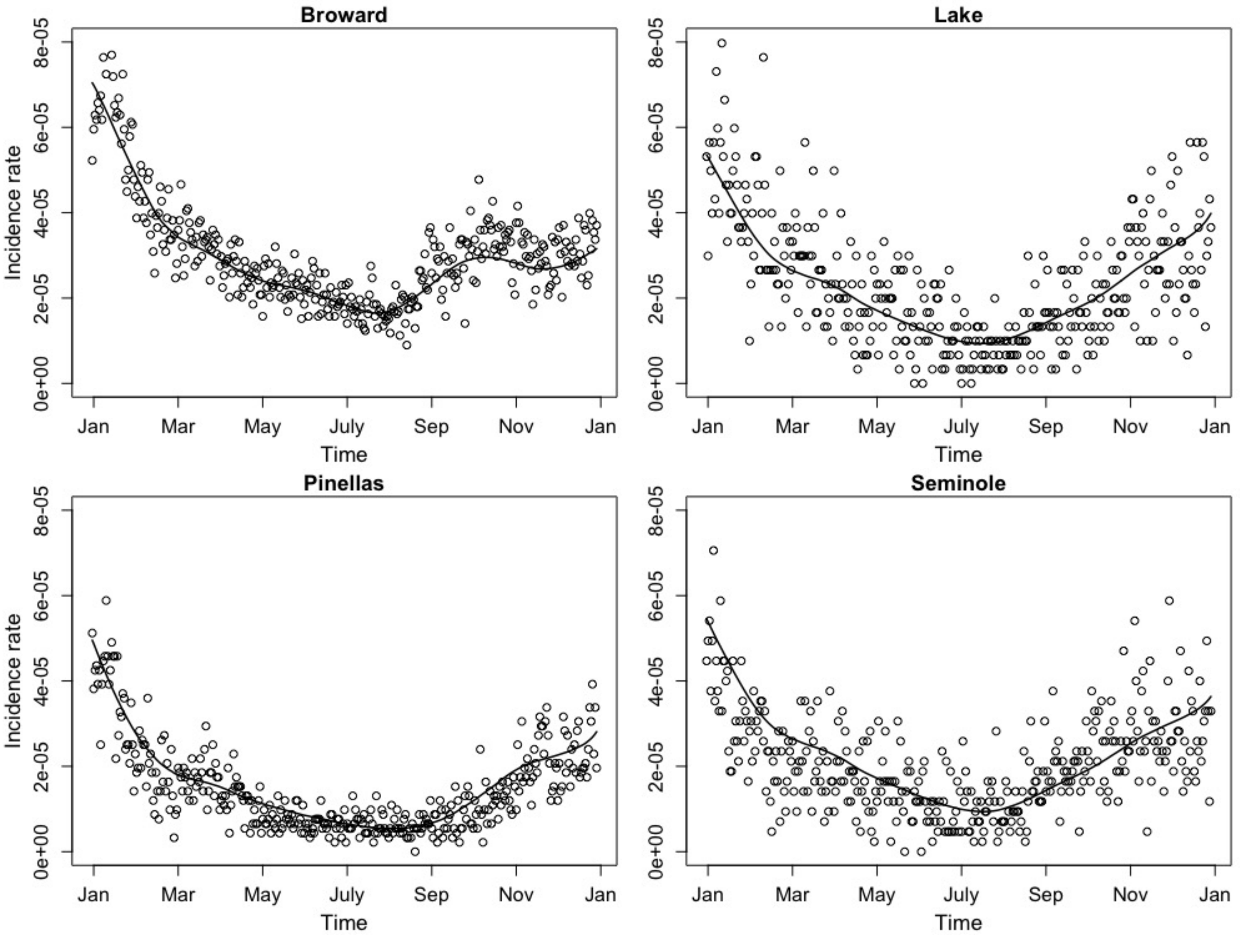

Figure 2: Observed ILI incidence rates in four Florida counties during the year 2013 (small circles) and the corresponding estimated mean functions (solid lines).

3.2 Spatio-temporal data monitoring

This subsection demonstrates how to monitor a spatio-temporal process by the methods described in Subsection 2.2 using the package **SpTe2M**. Recall that construction of the CUSUM chart (8) consists of two main steps. The first step is to model an IC spatio-temporal dataset and estimate the regular spatio-temporal pattern of the IC data. Then, in the second step, the control limit of the chart is first determined by the block bootstrap procedure and then the chart can be used for online process monitoring. See the discussions in Section 2 for more details. For the ILI data, the observed ILI incidence rates in the years

Table 2: Arguments of the function spte_covest() for estimating $V(t, t'; \boldsymbol{s}, \boldsymbol{s}')$.

Argument	Description
gt	The bandwidth g_t used in (5) and (6). Its default value is NULL. If gt=NULL, then gt is chosen by (A3) of the supplementary file.
gs	The bandwidth g_s used in (5) and (6). Its default value is NULL. If gs=NULL, then gs is chosen by (A3).
stE1	A three-column matrix specifying $(\boldsymbol{s}, t)$. Its default value is NULL. In such a case, stE1=st.
stE2	A three-column matrix specifying $(\boldsymbol{s}', t')$. Its default value is NULL. In such a case, stE2=st.

2012 and 2013 are quite stable (cf., Figure 1). So, these data will be used as the IC dataset for setting up the CUSUM chart (8). The IC observations are divided into two parts: the IC data in the year 2013 are used for estimating the mean and covariance functions of the IC spatio-temporal model, and the data in the year 2012 are used for determining the control limit of the chart, as described in Section B of the supplementary file. Then, the chart (8) is used to monitor process observations in the year 2014. To this end, we first decorrelate the observed ILI incidence rates in the year 2014 and then apply the CUSUM chart (8) to the decorrelated data. All these procedures can be accomplished by using the function sptemnt_cusum(). The main arguments of this function are listed below:

- y: A vector of spatio-temporal observations of the response y.
- st: A 3-column matrix specifying the locations and times of the observations in y.
- type: A vector specifying the type of each observation, and type could be "Mnt", "IC1" or "IC2", where type="Mnt" implies that the related observation is for process monitoring, type="IC1" implies that the related observation is in the IC data for determining the control limit, and type="IC2" implies that the related observation is in the IC data for estimating the IC spatio-temporal mean and covariance functions. If there is a single IC dataset specified, then it will be used for both purposes. If no observations are specified as IC data, then an error will be returned.
- ARL0: The pre-specified IC average run length ARL_0. The default value is 200.

- gamma: The pre-specified allowance constant γ in the CUSUM chart (8). The default value is 0.1.
- B: Bootstrap sample size. The default value is 1,000.
- bs: The block size b of the block bootstrap procedure. The default value is 5.

The following code is an example to monitor the ILI data using the CUSUM chart (8):

```
R> y <- ili_dat$Rate; st <- ili_dat[,c('Lat','Long','Time')]
R> type <- rep(c('IC1','IC2','Mnt'),c(N1,N2,N2))
R> ili.cusum <- sptemnt_cusum(y,st,type,ht=0.05,hs=6.5,gt=0.25,gs=1.5)
```

In this example, the bandwidths (h_t, h_s) for estimating the spatio-temporal mean function are chosen based on the modified cross-validation procedure in (A1)-(A2) of the supplementary file. This bandwidth selection procedure can be implemented by using the function mod_cv() in **SpTe2M**, and the selected bandwidths for (h_t, h_s) are 0.05 and 6.50, respectively. Regarding the bandwidths (g_t, g_s) for estimating the covariance function, they are determined by the mean square prediction error criterion defined in (A3) of the supplementary material. After running the function cv_mspe(), they are chosen to be 0.25 and 1.50, respectively, in this example.

The function sptemnt_cusum() in the above code returns a list, and the computed charting statistic values and the control limit of the CUSUM chart (8) can be accessed via ili.cusum$cstat and ili.cusum$cl, respectively. Then, we can plot the CUSUM charting statistic values versus the observation times by using the following code:

```
# plot the CUSUM chart
R> cstat <- ili.cusum$cstat; cl <- ili.cusum$cl
R> plot(1:365,cstat,type="l"); abline(h=cl,lty=2)
```

The resulting plot is shown in the left panel of Figure 3. To better perceive the charting statistic values around its first signal time 10/16/2014, the charting statistic values during the time period from 09/15/2014 to 10/31/2014 are presented in the right panel of Figure 3.

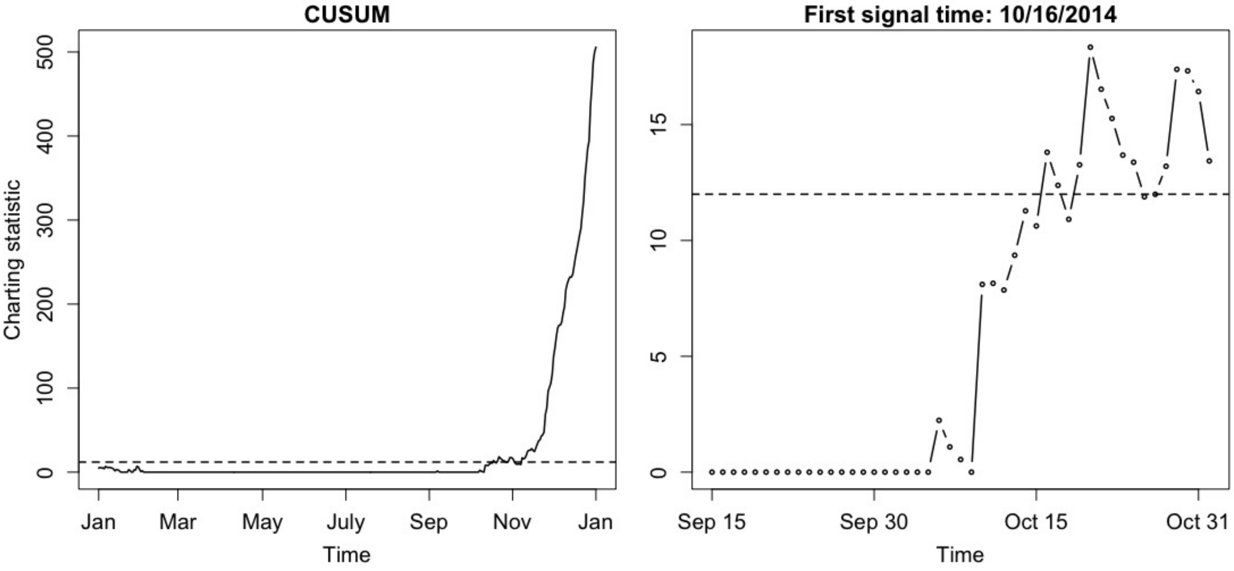

Figure 3: The CUSUM chart for monitoring the ILI data in the year 2014 (left panel), together with its zoom-in part during the time period from 09/15–10/31 in 2014 (right panel). In each plot, the horizontal line denotes the control limit of the chart.

In some applications, we may want to apply the data decorrelation and standardization procedure to some spatio-temporal observations. This can be accomplished by using the spte_decor() function. For instance, to decorrelate the observed ILI incidence rates in the year 2013, we can do the following:

```
R> decor <- spte_decor(y=y.sub,st=st.sub,y0=y.sub,st0=st.sub)
```

Then, the decorrelated and standardized data can be obtained via decor$std.res. The arguments y, st, y0 and st0 of the above R command are explained below:

- y: A vector of the spatio-temporal observations to decorrelate.
- st: A 3-column matrix specifying the locations and times of the observations in y.

- y0: A vector of the IC spatio-temporal observations that we can use to estimate the IC mean and covariance functions by the procedures (3)-(4) and (5)-(6).

- st0: A 3-column matrix specifying the locations and times of the observations in y0.

In the ILI dataset, there are observations of the two covariates “air temperature” and “relative humidity” available. Next, we try to monitor the observed ILI incidence rates data by using these covariate information. To this end, the association between the two covariates and the ILI incidence rates can be investigated by fitting the semiparametric spatio-temporal model (9), which can be accomplished by using spte_semiparmreg(). The arguments y, st, ht, hs and stE of this function have the same meaning as those of the function spte_meanest() discussed earlier. The other arguments of spte_semiparmreg() are explained below:

- x: A p-column matrix containing the observed data of the p covariates.

- maxIter: The maximum number of iterations allowed in the iterative estimation procedure for estimating Model (9). Its default value is 1,000.

- tol: The tolerance level ς of the convergence criterion in the iterative estimation procedure for estimating Model (9). Its default value is 0.0001.

Then, we can use the following code to fit the semiparametric model for the ILI data in 2013, and the estimated regression coefficients $\widehat{\boldsymbol{\beta}}$ can be obtained via semi.est$beta. From the estimated model, the estimated regression coefficients of “relative humidity” and “air temperature” are -1.17×10^{-6} and -1.06×10^{-6}, respectively. Therefore, both covariates are negatively associated with the observed ILI incidence rates. This is consistent with our intuition and the conclusions found in the literature (cf., Pica and Bouvier 2012).

```
R> x.sub  <- as.matrix(subdat[,c('Temp','RH')])
R> semi.est <- spte_semiparmreg(y=y.sub,st=st.sub,x=x.sub)
```

After the semiparametric model (9) is estimated, the EWMAC chart (10) can be used to monitor the observed ILI incidence rates from the beginning of the year 2014. This can be achieved by using the function sptemnt_ewmac() as follows:

```
R> x <- as.matrix(ili_dat[,c('Temp','RH')])
R> ili.ewmac <- sptemnt_ewmac(y,x,st,type,ht=0.05,hs=6.5,gt=0.25,gs=1.5)
# plot the EWMAC chart
R> cstat <- ili.ewmac$cstat; cl <- ili.ewmac$cl
R> plot(1:365,cstat,type="l"); abline(h=cl,lty=2)
```

Most arguments of sptemnt_ewmac() are the same as those of sptemnt_cusum(). In the function sptemnt_ewmac(), ARL0.z is the pre-specified IC ARL value for the covariate chart $\{E_{z,i}, i \geq 1\}$ (cf., the related discussion in Section C of the supplementary file) whose default value is 200, and lambda is the weighting parameter λ used in the EWMAC chart (10) whose default value is 0.1. The computed charting statistic values and the control limit of the EWMAC chart (10) can be obtained using ili.ewmac$cstat and ili.ewmac$cl, respectively. Figure 4 shows the EWMAC chart for monitoring the ILI data in the year 2014 (left panel), as well as its zoom-in part in the time period from 09/25/2014 to 10/31/2014 (right panel). From Figure 4, it is clear that the EWMAC chart gives its first signal on 09/23/2014 which is 23 days earlier than the first signal of the CUSUM chart (8). So, it is indeed beneficial to use the covariate information for monitoring the ILI incidence rates.

Finally, we demonstrate the application of the EWSL chart (12) by using the function sptemnt_ewsl(). To this end, let us use the following code:

```
R> ili.ewsl <- sptemnt_ewsl(y,st,type,ht=0.05,hs=6.5,gt=0.25,gs=1.5)
# plot the EWSL chart
R> cstat <- ili.ewsl$cstat; cl <- ili.ewsl$cl
R> plot(1:365,cstat,type="l"); abline(h=cl,lty=2)
```

As can be seen from the code, the arguments of the function sptemnt_ewsl() are the

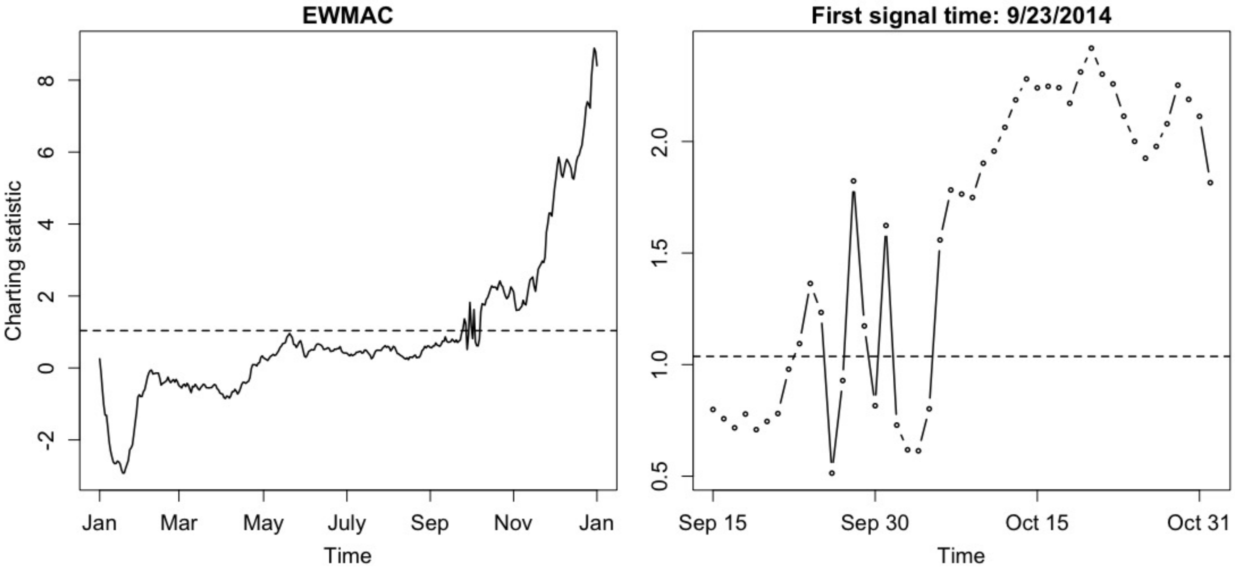

Figure 4: The EWMAC chart for monitoring the ILI data in the year 2014 (left panel), together with its zoom-in part in the time period from 09/25/2014 to 10/31/2014 (right panel). In each plot, the horizontal line denotes the control limit of the chart.

same as those of the function sptemnt_ewmac(). Here, we would like to mention that the EWSL chart is based on the penalized exponentially weighted kernel smoothing procedure (11). To use this procedure, the bandwidth h and the tuning parameters (ψ_1, ψ_2) should be chosen properly. When using sptemnt_ewsl(), h is chosen by the cross-validation procedure in (A6) and (ψ_1, ψ_2) are chosen by BIC defined in (A7) of the supplementary file.

To plot the EWSL chart, we can first obtain the EWSL charting statistic values and the control limit via ili.ewsl$cstat and ili.ewsl$cl. Then, the plot can be generated, which is shown in the left panel of Figure 5. Its zoom-in part in the time period during 09/15/2014-10/31/2014 is shown in the right panel of the figure. From the figure, we can see that the EWSL chart gives its first signal on 10/6/2014, which is 10 days earlier than that of the CUSUM chart (8) shown in Figure 3.

It should be pointed out that we choose to use the county centroids as the locations of the 67 Florida counties in this example because this strategy is commonly used in the statistical

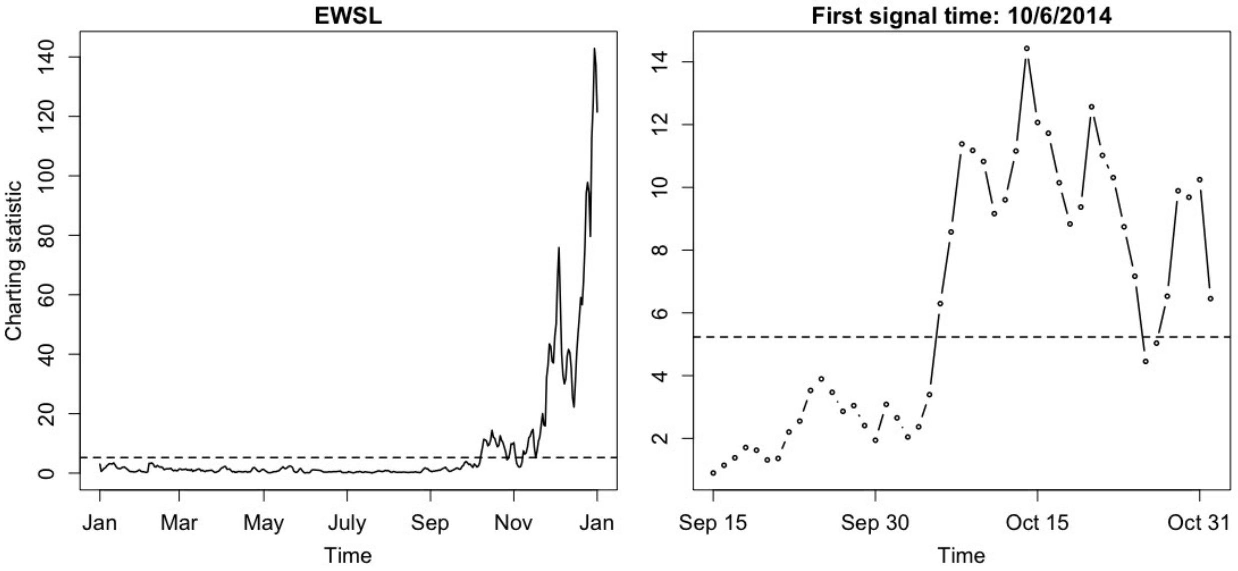

Figure 5: The EWSL chart for monitoring the ILI data in the year 2014 (left panel), together with its zoom-in part in the time period during 09/15/2014-10/31/2014 (right panel). In each plot, the horizontal line denotes the control limit of the chart.

and epidemiological literatures (e.g., Hooten et al. 2019). In some applications, however, it may be more natural to represent locations of the areal units by population centers or some other strategies. For instance, most disease cases occur around the population centers rather than the county centroids for some counties in the ILI example. In such a case, population centers might be more appropriate to represent the county locations. After taking this into consideration, we also implement the related methods in the package by using the population centers as the county locations. The corresponding results are presented in Figure S1 and Table S1 of the supplementary file. By comparing these results with those described above, it can be seen that the performance of our methods is quite robust to the choice of county locations, which can be explained below. Our spatio-temporal data modeling methods are based on the local kernel smoothing procedure. The related nonparametric estimators at a given location and time are weighted averages of neighboring data, where the weights are determined by the kernel function and the neighborhood size is controlled by the bandwidths. After replacing the county centroids by the population centers as the county locations, the weights in our estimation procedures become different for some counties but the selected bandwidths are almost the same. In the kernel smoothing literature, it has been well discussed that the choice of the kernel functions (or the weights) is less important than the choice of the bandwidths (cf., Brabanter et al. 2011). This explains why our estimators are quite robust to the choice of the county locations in this example.

4 PM2.5 Data Example

In this section, we further demonstrate the main functions in **SpTe2M** by using another real data example related to the observed PM2.5 concentration levels described below. PM2.5 refers to fine inhalable particles in the air, with diameters less than 2.5 micrometers. Since PM2.5 concentration is positively associated with the incidence of many diseases such as cancers and respiratory diseases, governments around the world have invested a great amount of money in establishing data collection and monitoring systems to collect PM2.5 data and monitor them sequentially over time. For instance, the China National Environmental Monitoring Centre (CNEMC) has developed such a system to collect daily PM2.5 concentrations at 183 major cities in China. Next, we apply the SpTe2M package to the PM2.5 data collected by CNEMC during the years 2014-2016, which contain the observed data collected at 183 air pollution monitoring stations. Unlike the ILI dataset, this dataset corresponds to a continuous geospatial process in the space domain. We choose to apply the methods in SpTe2M to the ILI and PM2.5 datasets to illustrate that they are useful for analyzing both areal and geospatial data.

Let us first investigate the spatio-temporal pattern of the PM2.5 data by estimating its mean and covariance functions. To this end, we first use the functions mod_cv() and cv_mspe() to select the bandwidths (h_t, h_s) and (g_t, g_s) that are used in the procedures (3)-(4) and (5)-(6), respectively.

```
R> data(pm25_dat); n <- 365; m <- 183
```

```
R> pm.sub <- pm25_dat[1:(n*m),] # Extract the observed data in the year 2014
R> y.sub <- pm.sub$PM2.5; st.sub <- pm.sub[,c('Lat','Long','Time')]
R> mcv <- mod_cv(y=y.sub,st=st.sub)
R> mcv$bandwidth.opt
[1] 0.15    11.21
R> mspe <- cv_mspe(y=y.sub,st=st.sub)
R> mspe$bandwidth.opt
[2] 0.74    10.54
```

Then, these selected bandwidths are used in the local linear kernel smoothing procedure (3)-(4) and the weighted moment estimation procedure (5)-(6) when applying the functions spte_meanest() and spte_covest() to the PM2.5 data in the year 2014.

```
R> mu.est <- spte_meanest(y=y.sub,st=st.sub,ht=0.15,hs=11.21)
R> cov.est <- spte_covest(y=y.sub,st=st.sub,gt=0.74,gs=10.54)
```

The observed PM2.5 concentration levels along with the estimated mean values for the four representative cities in China: Luzhou, Meizhou, Panzhihua and Shenzhen are presented in Figure 6, which is made using the following code. It can be seen from the figure that the estimated mean values describe the observed data well.

```
R> mu <- mu.est$muhat; mu <- t(matrix(mu,m,n)); obs <- t(matrix(y.sub,m,n))
R> id <- c(87,90,100,121); par(mfrow=c(2,2))
R> plot(1:365,mu[,id[1]],type='l',lty=1,lwd=1.5); points(1:365,obs[,id[1]])
R> plot(1:365,mu[,id[2]],type='l',lty=1,lwd=1.5); points(1:365,obs[,id[2]])
R> plot(1:365,mu[,id[3]],type='l',lty=1,lwd=1.5); points(1:365,obs[,id[3]])
R> plot(1:365,mu[,id[4]],type='l',lty=1,lwd=1.5); points(1:365,obs[,id[4]])
```

Next, we will discuss how to monitor the observed PM2.5 data by using the functions sptemnt_cusum() and sptemnt_ewsl(). We cannot use the function sptemnt_ewmac() in this

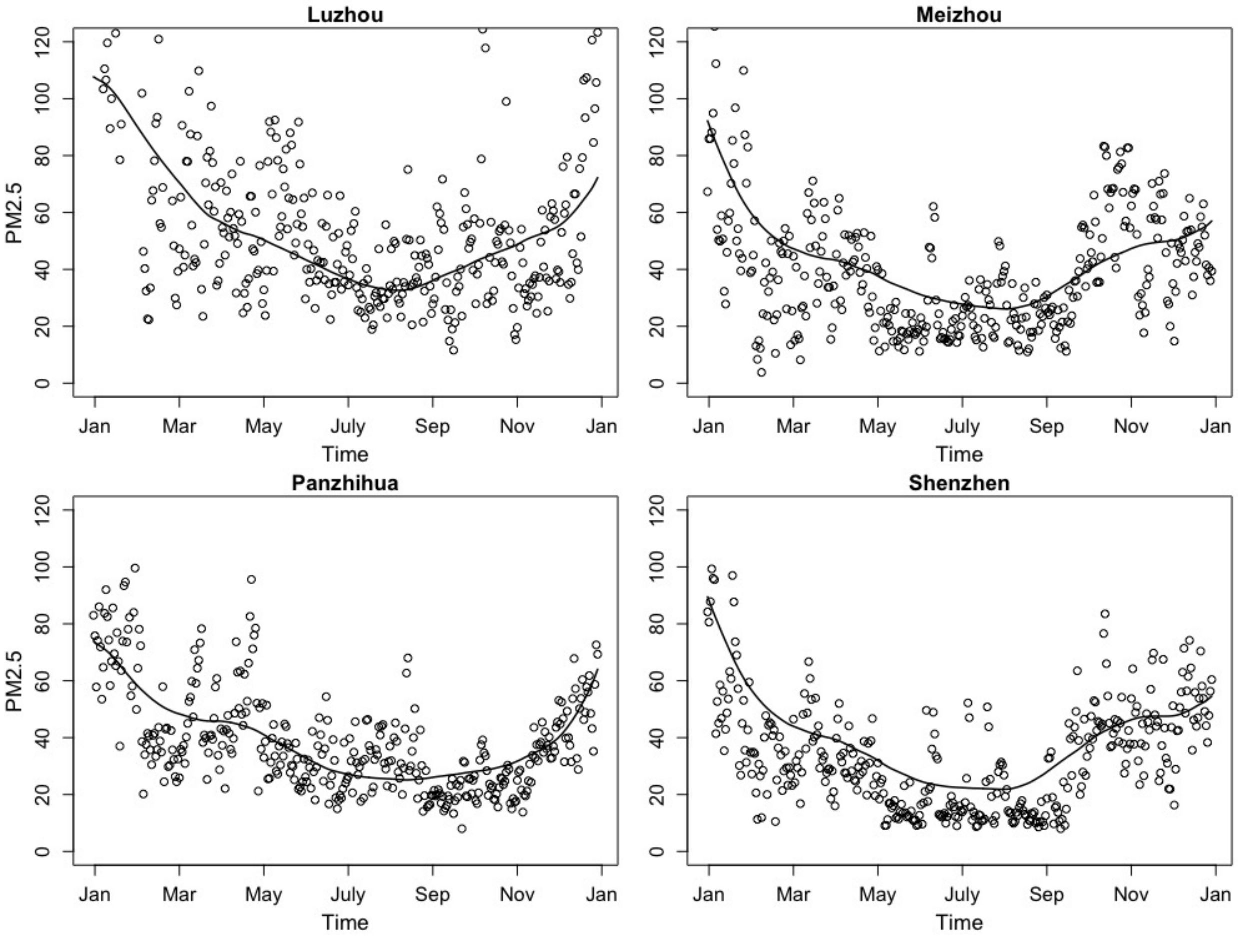

Figure 6: Observed PM2.5 concentration levels at four cities in China during the year 2014 (small circles) and the corresponding estimated mean functions (solid lines).

example since there are no covariates contained in the PM2.5 dataset. To use the related functions, we first need to specify the argument `type`, which separates the observed data into three parts: the IC observations for estimating the IC spatio-temporal model, the IC observations for determining the control limit of a chart, and the observations for process monitoring. To this end, the observed data in the year 2014 are used for estimating the IC model, the observed data in the year 2015 are used for determining the control limits of the related control charts, and the observed data in the year 2016 are for online process monitoring using the CUSUM chart (8) and the EWSL chart (12). The specific code for computing the charting statistic values and making the related control plots is given below.

```
R> y <- pm25_dat$PM2.5; st <- pm25_dat[,c('Lat','Long','Time')]
R> type <- rep(c('IC2','IC1','Mnt'),each=(n*m))
## Compute the CUSUM charting statistic values
R> pm.cusum <- sptemnt_cusum(y,st,type,ht=0.15,hs=11.21,gt=0.74,gs=10.54)
# Plot the CUSUM chart
R> cstat <- pm.cusum$cstat; cl <- pm.cusum$cl
R> plot(1:365,cstat,type="l"); abline(h=cl,lty=2)
## Compute the EWSL charting statistic values
R> pm.ewsl <- sptemnt_ewsl(y,st,type,ht=0.15,hs=11.21,gt=0.74,gs=10.54)
# Plot the EWSL chart
R> cstat <- pm.ewsl$cstat; cl <- pm.ewsl$cl
R> plot(1:365,cstat,type="l"); abline(h=cl,lty=2)
```

The two control charts are presented in Figure 7. From the figure, it can be seen that the first signals of the CUSUM chart (8) and the EWSL chart (12) are given on 4/7/2016 and 3/17/2016, respectively. So, in this example, the first signal of the EWSL chart is about 20 days earlier than that of the CUSUM chart.

5 Concluding Remarks

This paper introduces the R package **SpTe2M** that was developed recently to implement some nonparametric methods for modeling and monitoring spatio-temporal data discussed in Yang and Qiu (2018, 2019, 2020, 2022) and Qiu and Yang (2021, 2023). These methods can well accommodate the complex structures of spatio-temporal processes, and thus are reliable to use in practice. Some original computer codes of these methods were written in Fortran for efficient computation. For users' convenience, **SpTe2M** has packaged these Fortran codes in R. Thus, the package **SpTe2M** should provide a convenient and effective tool for modeling and monitoring spatio-temporal data. However, there are still some issues about the package

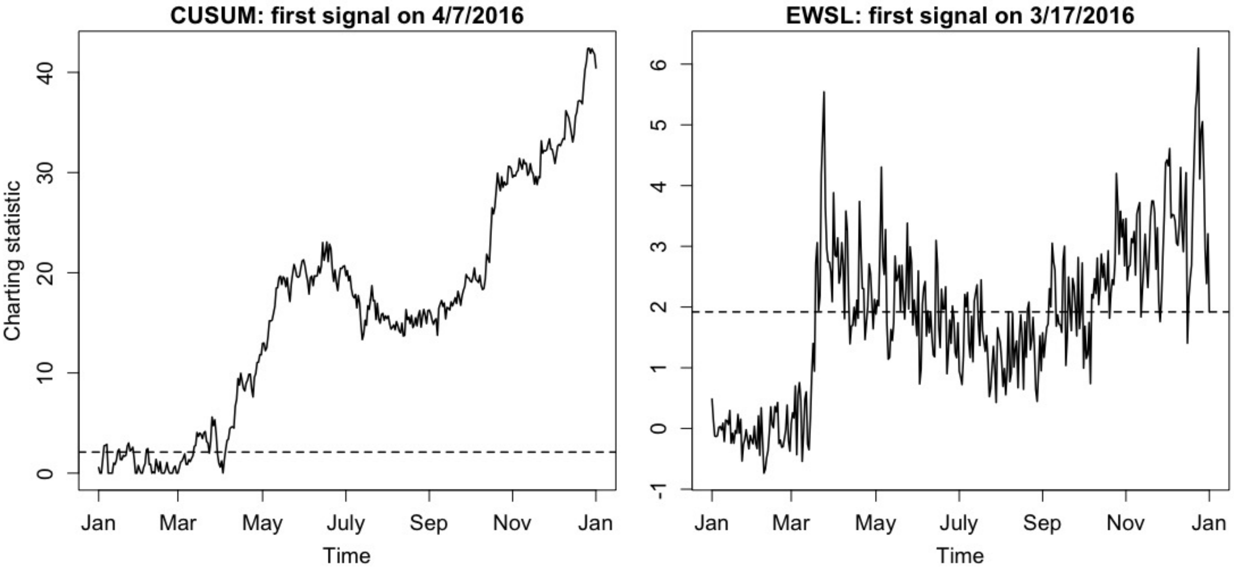

Figure 7: The two control charts CUSUM and EWSL for monitoring the observed PM2.5 concentration levels in the year 2016. The dashed horizontal line in each plot denotes the control limit of the related control chart.

that need to be addressed in the future. For example, the current package can only handle cases with univariate spatio-temporal data (i.e., the response variable $y(t, \boldsymbol{s})$ is univariate). It is our belief that the related methods can be extended to cases with multiple spatio-temporal response variables. In addition, it is often our interest in practice to study the relationship between the response variable $y(t, \boldsymbol{s})$ and some of its covariates, as discussed in Section 2. The semiparametric model (9) and the related R function spte_semiparmreg() in the package are designed for such a purpose. But, they depend on the assumption that the relationship is linear. Of course, this assumption could be violated in certain applications. In such cases, the semiparametric model (9) should be generalized to the one with space/time-varying regression coefficients or even to a nonparametric model. In addition, after the related control charts give their signals, a reliable post-signal diagnostic tool should be developed to figure out when and where the detected shifts occur in both the time and space domains. All these issues will be addressed in our future research, and the R package **SpTe2M** will be updated accordingly.

Acknowledgments: The authors would like to thank the editor, the associate editor, and two referees for their insightful comments and suggestions, which improved the quality of the paper greatly.

Data availability statement: The Florida ILI data and PM2.5 concentration data are included in **SpTe2M**. You can have access to them after you download and install the package from CRAN.

Funding: This work was supported in part by a grant from the Clinical and Translational Science Institute under grant number UL1TR001436.

References

Allévius, B. (2018), "Scanstatistics: space-time anomaly detection using scan statistics," *Journal of Open Source Software*, **3**, 515.

Apley, D.W., and Tsung, F. (2002), "The autoregressive T^2 chart for monitoring univariate autocorrelated processes," *Journal of Quality Technology*, **34**, 80–96.

Bakar, K.S., and Sahu, S.K. (2015), "spTimer: spatio-temporal Bayesian modeling using R," *Journal of Statistical Software*, **63**, 1–32.

Brabanter, K.D., Brabanter, J.D., Suykens, J.A.K., and De Moor, B. (2011), "Kernel regression in the presence of correlated errors," *Journal of Machine Learning Research*, **12**, 1955–1976.

Bradley, J.R., Holan, S.H., and Wikle, C.K. (2015), "Multivariate spatio-temporal models for high-dimensional areal data with application to longitudinal employer-household dynamics," *Annals of Applied Statistics*, **9**, 1761–1791.

Cressie, N., and Wikle, C. (2011), *Statistics for Spatio-Temporal Data*, New York: Wiley.

Diggle, P.J., Moraga, P., Rowlingson, B., and Taylor, B.M. (2013), "Spatial and spatio-temporal log-Gaussian Cox processes: extending the geostatistical paradigm," *Statistical Science*, **28**, 542–563.

Epanechnikov, V.A. (1969), "Non-parametric estimation of a multivariate probability density," *Theory of Probability and its Applications*, **14**, 153–158.

Finley, A.O., Banerjee, S., and Gelfand, A. (2015), "SpBayes for large univariate and multivariate point-referenced spatio-temporal data models," *Journal of Statistical Software*, **63**, 1–28.

Finley, A.O., Datta, A., and Banerjee, S. (2020), "R package for nearest neighbor Gaussian process models," *arXiv:2001.09111.*

Fiore A.E., Uyeki, T.M., Broder, K., Finelli, L., Euler, G.L., Singleton, J.A., Iskander, J.K.,Wortley, P.M., Shay, D.K., Bresee, J.S., Cox, N.J. (2010), "Prevention and control of influenza with vaccines: recommendations of the advisory committee on immunization practices," *MMWR Recommendations and Reports*, **59**, 1–62.

Gramacy, R.B. (2016), "laGP: large-scale spatial modeling via local approximate Gaussian processes in R," *Journal of Statistical Software*, **72**, 1–46.

Guhaniyogi, R., and Banerjee, S. (2018), "Meta-kriging: scalable Bayesian modeling and inference for massive spatial datasets," *Technometrics*, **60**, 430–444.

Hooten, M.B., Ver Hoef, J.M., and Hanks, E.M. (2019), "Simultaneous autoregressive (SAR) model," *Wiley StatsRef: Statistics Reference Online*,
DOI: 10.1002/9781118445112.stat08208.

Meyer, S., Held, L., and Höhle, M. (2017), "Spatio-temporal analysis of epidemic phenomena using the R package surveillance," *Journal of Statistical Software*, **77**, 1–55.

Møller, J., Syversveen, A.R., and Waagepetersen, R.P. (1998), "Log-gaussian cox processes," *Scandinavian Journal of Statistics*, **25**, 451–482.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015), "A multiresolution Gaussian process model for the analysis of large spatial datasets," *Journal of Computational and Graphical Statistics*, **24**, 579–599.

Pica, N., Bouvier, N.M. (2012), "Environmental factors affecting the transmission of respiratory viruses," *Current Opinion in Virology*, **2**, 90–95.

Qiu, P. (2014), *Introduction to Statistical Process Control*, Boca Raton, FL: CRC Press.

Qiu, P., Li, W., and Li, J. (2020), "A new process control chart for monitoring short-range serially correlated data," *Technometrics*, **62**, 71–83.

Qiu, P., and Yang, K. (2021), "Effective disease surveillance by using covariate information," *Statistics in Medicine*, **40**, 5725–5745.

Qiu, P., and Yang, K. (2023), "Spatio-temporal process monitoring using exponentially weighted spatial LASSO," *Journal of Quality Technology*, **55**, 163–180.

Risser, M.D., and Turek, D. (2020), "Bayesian inference for high-dimensional nonstationary Gaussian processes," *Journal of Statistical Computation and Simulation*, **90**, 2902–2928.

Stroud, J.R., Müller, P., and Sansó, B. (2001), "Dynamic models for spatiotemporal data," *Journal of the Royal Statistical Society: Series B*, **63**, 673–689.

Taylor, B.M., Davies, T.M., Rowlingson, B.S., and Diggle, P.J. (2013), "lgcp: An R package for inference with spatial and spatio-temporal log-Gaussian Cox processes," *Journal of Statistical Software*, **52**, 1–40.

Wang, A., Xian, X., Tsung, F., and Liu, K. (2018), "A spatial-adaptive sampling procedure for online monitoring of big data streams," *Journal of Quality Technology*, **50**, 329–343.

Wikle, C.K., and Hooten, M.B. (2010), "A general science-based framework for dynamical spatio-temporal models," *Test*, **19**, 417–451.

Yang, K., and Qiu, P. (2018), “Spatio-temporal incidence rate data analysis by nonparametric regression,” *Statistics in Medicine*, **37**, 2094–2107.

Yang, K., and Qiu, P. (2019), “Nonparametric estimation of the spatio-temporal covariance structure,” *Statistics in Medicine*, **38**, 4555–4565.

Yang, K., and Qiu, P. (2020), “Online sequential monitoring of spatio-temporal disease incidence rates,” *IISE Transactions*, **52**, 1218–1233.

Yang, K., and Qiu, P. (2022), “A three-step local smoothing approach for estimating the mean and covariance functions of spatio-temporal data,” *Annals of the Institute of Statistical Mathematics*, **74**, 49–68.

Zammit-Mangion, A., and Cressie, N. (2017), “FRK: an R package for spatial and spatio-temporal prediction with large datasets,” *arXiv:1705.08105*.

Zou, H. (2006), “The adaptive LASSO and its oracle properties,” *Journal of the American Statistical Association*, **101**, 1418–1429.