



Modelling Big, Heterogeneous, Non-Gaussian Spatial and Spatio-Temporal Data using **FRK**

Matthew Sainsbury-Dale Andrew Zammit-Mangion Noel Cressie
University of Wollongong University of Wollongong University of Wollongong

Abstract

Non-Gaussian spatial and spatio-temporal data are becoming increasingly prevalent, and their analysis is needed in a variety of disciplines. **FRK** is an R package for spatial/spatio-temporal modelling and prediction with very large data sets that, to date, has only supported linear process models and Gaussian data models. In this paper, we describe a major upgrade to **FRK** that allows for non-Gaussian data to be analysed in a generalised linear mixed model framework. These vastly more general spatial and spatio-temporal models are fitted using the Laplace approximation via the software **TMB**. The existing functionality of **FRK** is retained with this advance into non-Gaussian models; in particular, it allows for automatic basis-function construction, it can handle both point-referenced and areal data simultaneously, and it can predict process values at any spatial support from these data. This new version of **FRK** also allows for the use of a large number of basis functions when modelling the spatial process, and is thus often able to achieve more accurate predictions than previous versions of the package in a Gaussian setting. We demonstrate innovative features in this new version of **FRK**, highlight its ease of use, and compare it to alternative packages using both simulated and real data sets.

Keywords: areal data, basis functions, big data, change-of-support, fixed rank kriging, non-Gaussian data, spatial statistics.

1. Introduction

Non-Gaussian spatial and spatio-temporal data arise from a vast array of sources, and the statistical modelling of these data is pertinent, as accurate predictions, and uncertainty quantification of those predictions, give informed answers to real-world problems.

There are, by now, several approaches to statistical modelling and spatial/spatio-temporal prediction with non-Gaussian data. One widespread method to deal with non-Gaussian data is *trans-Gaussian kriging* (Cressie 1993, pg. 137–138), in which standard kriging (i.e., spa-

tial optimal linear prediction) is used after applying a non-linear transformation to the data, and approximately unbiased predictions are made on the original scale using a delta-method approximation. Several other approaches hinge on the use of a spatial version of the generalised linear mixed model (GLMM), whereby the response distribution is assumed to be a member of the exponential family of distributions (e.g., McCullagh and Nelder 1989), and the mean is modelled using a transformation of some latent spatial process $Y(\cdot)$ (Diggle, Tawn, and Moyeed 1998). In their seminal work, Diggle *et al.* (1998) employed a stationary model for $Y(\cdot)$ within the spatial GLMM framework, and a Markov chain Monte Carlo (MCMC) algorithm to obtain predictive distributions. Optimal prediction or estimation of unknown quantities from m observations entails the inversion of an $m \times m$ covariance matrix for many statistical models. Since this task is generally $O(m^3)$ in computational complexity, alternative approaches that scale well with sample size, which we collectively refer to as *fixed-rank* approaches, are often employed in ‘big data’ settings.

Fixed-rank variants of trans-Gaussian kriging are relatively under-developed (see Cressie, Sainsbury-Dale, and Zammit-Mangion 2021, sec. 3.2.1), however many modellers have used fixed-rank variants of the spatial GLMM. A popular fixed-rank model for $Y(\cdot)$ is the so-called spatial random effects (SRE) model, where $Y(\cdot)$ is modelled as a linear combination of a fixed number of spatial basis functions with spatially correlated random coefficients (Cressie and Johannesson 2008): For example, Sengupta and Cressie (2013) and Bradley, Wikle, and Holan (2016) use it in the spatial GLMM context. Lindgren, Rue, and Lindström (2011) modelled $Y(\cdot)$ by linking Gaussian fields (GFs) with Gaussian Markov random fields (GMRFs) via stochastic partial differential equations (SPDEs), with dimension-reduction facilitated by the finite-element method. Finley, Datta, and Banerjee (2020) modelled binomial data using a spatial GLMM with $Y(\cdot)$ a nearest neighbour Gaussian process (NNGP; Datta, Banerjee, Finley, and Gelfand 2016). Lee and Park (2020) partitioned the spatial domain into disjoint subregions and, for each subregion, a spatial GLMM model was used independently of the other subregions. Then the global process was constructed as a weighted sum of the mutually independent local processes. The fixed-rank spatial GLMM naturally extends to the spatio-temporal setting; see, for example, Lopes, Gamerman, and Salazar (2011), Bradley, Holan, and Wikle (2018), Bradley, Wikle, and Holan (2019), and Zhang and Cressie (2020).

Despite the many modelling approaches available, software for spatial and spatio-temporal modelling of non-Gaussian data (for which classical kriging-based approaches are linear and hence may be sub-optimal) are relatively limited. Software dedicated to this task include the R (R Core Team 2021) packages **ngspatial** (Hughes 2014), **spBayes** (Finley, Banerjee, and Gelfand 2015), **mgcv** (Wood 2017), **spNNGP** (Finley *et al.* 2020), **georob** (Papritz 2020), and **spatialfusion** (Wang and Furrer 2021). Each of these packages has a different set of limitations: **spBayes**, **mgcv**, and **spNNGP** are limited to point-referenced data; **spBayes** uses basis functions that depend on covariance-function parameters, so that computationally it can only handle a small number of predictive-process knots, which in turn yields a high degree of smoothing; **georob** is not designed for large data sets; and **ngspatial**, **spBayes**, **spNNGP**, **georob**, and **spatialfusion** are restricted to the spatial setting, where they cater for only a small number of non-Gaussian distributions. Further, with the exception of **spatialfusion**, these software packages do not cater for spatial change-of-support. Some general-purpose packages (e.g., **INLA**; Rue, Martino, and Chopin 2009; Lindgren and Rue 2015) can, in principle, handle the wide array of modelling challenges posed by non-Gaussian spatial and spatio-temporal data; however, they are not specifically designed for this purpose and can be

difficult for an unfamiliar user to implement. The package **inlabru** (Bachl, Lindgren, Borchers, and Illian 2019) aims to facilitate Gaussian and non-Gaussian spatial modelling using **INLA** but, at the time of writing, it does not implement spatio-temporal modelling.

FRK (Zammit-Mangion and Cressie 2021) is an R package for spatial/spatio-temporal statistical modelling and prediction. In this article, we present a major upgrade to **FRK** that allows one to cater for many distributions within the exponential family using the spatial GLMM framework; we henceforth refer to it as **FRK** v2 and the original version as **FRK** v1. **FRK** v2 provides a unifying framework that handles large, spatial and spatio-temporal non-Gaussian (and Gaussian) data, and it can seamlessly ingest point-referenced and area-referenced data to solve spatial change-of-support problems. User-friendliness is a central focus of the package: Challenging statistical analyses may be tackled with only a few lines of intuitive, readable code. Optimal spatial prediction proceeds through the use of an empirical hierarchical statistical model (where likelihood-based estimates are substituted in place of unknown parameters) and a Monte Carlo (MC) algorithm, where a minimal number of user-level decisions is required. **FRK** v2 also accommodates the modelling of non-Gaussian spatial and spatio-temporal data on the surface of a sphere, a feature not offered by many other packages. Finally, although the primary motivation for this major upgrade is the modelling of non-Gaussian data, **FRK** v2 also allows for the use of substantially more basis functions than **FRK** v1, which often results in more accurate predictions when in a Gaussian setting.

The remainder of the paper is organised as follows. In Section 2, we establish the statistical framework for **FRK** v2, and we describe model fitting and prediction. In Section 3, we discuss and illustrate the new functionalities in **FRK** v2. In Section 4, we present a comparative study between **FRK** v2 and several related packages, as well as real-world applications of **FRK** v2. Section 5 gives a discussion and conclusions.

2. Methodology

The model used in **FRK** v2 is a spatial or spatio-temporal hierarchical statistical model consisting of two conditional-probability layers. In the *process layer*, we model the conditional mean of the data as a transformation of a latent spatial process modelled as a low-rank SRE model; see Section 2.1. In the *data layer*, we use a conditionally independent exponential-family model for each element of the data vector; see Section 2.2. In Section 2.3, we discuss parameter estimation and, in Section 2.4, we discuss spatial prediction and uncertainty quantification of the predictions. In Section 2.5, we consider two distributions that have an assumed-known ‘size’ parameter, namely the binomial distribution and the negative-binomial distribution. In Section 2.6, we present the approach of **FRK** v2 for spatio-temporal data.

2.1. The process layer

The process layer, which governs the conditional mean of the data, retains many similarities to that in **FRK** v1. Note that here we discuss the spatial case only; the extension to a spatio-temporal setting is outlined in Section 2.6.

We denote the latent spatial process as $Y(\cdot) \equiv \{Y(\mathbf{s}): \mathbf{s} \in D\}$, where \mathbf{s} indexes space in the spatial domain of interest D . The model for the latent process is

$$Y(\mathbf{s}) = \mathbf{t}(\mathbf{s})^\top \boldsymbol{\alpha} + v(\mathbf{s}) + \xi(\mathbf{s}); \quad \mathbf{s} \in D, \tag{1}$$

where each term in (1) models a different type of spatial variability. First, spatially referenced covariates $\mathbf{t}(\cdot)$ and their associated regression parameters $\boldsymbol{\alpha}$, capture spatial variation that is linked to known, usually large-scale, explanatory variables that are elements of $\mathbf{t}(\cdot)$; the model requires that the covariates are known at every location in D . Second, the spatially correlated random effect $v(\cdot)$ captures medium-to-small-scale spatial variation. If the spatial process being modelled has fine-scale variation, including only $\mathbf{t}(\cdot)^\top \boldsymbol{\alpha}$ and $v(\cdot)$ can result in an overly smooth spatial model and hence overly optimistic predictions. This problem is alleviated by also including a fine-scale-variation random process, $\xi(\cdot)$, in the model.

In **FRK**, the medium-to-small-scale term $v(\cdot)$ is constructed as a linear combination of r spatial basis functions with random coefficients, where r is fixed and usually smaller than m , the number of observations. Specifically,

$$v(\mathbf{s}) = \sum_{l=1}^r \phi_l(\mathbf{s}) \eta_l = \boldsymbol{\phi}(\mathbf{s})^\top \boldsymbol{\eta}; \quad \mathbf{s} \in D,$$

where $\boldsymbol{\eta} \equiv (\eta_1, \dots, \eta_r)^\top$ is an r -dimensional vector of random coefficients for the r -dimensional vector $\boldsymbol{\phi}(\cdot) \equiv (\phi_1(\cdot), \dots, \phi_r(\cdot))^\top$ of pre-specified spatial basis functions. See Appendix A for details on how these basis functions are constructed. The fine-scale term, $\xi(\cdot) \equiv \{\xi(\mathbf{s}) : \mathbf{s} \in D\}$, is modelled as white noise after discretisation, which we discuss next.

To cater for different observation supports and facilitate solutions to spatial change-of-support problems, **FRK** assumes a discretised domain of interest, $D^G \equiv \{A_i : i = 1, \dots, N\}$, that is made up of N non-overlapping basic areal units (BAUs) such that $D = \bigcup_{i=1}^N A_i$. The finest resolution at which one can make predictions is at the level of the BAU. Point predictions are not possible; however, when the BAUs are so fine that the chosen basis functions are approximately constant within each BAU, the BAU-level predictions are practically equivalent to point predictions (see [Zammit-Mangion and Cressie, 2021](#), Sec. 2, for more details). Now, let $Y(A_i)$ denote a representative value of $\{Y(\mathbf{s}) : \mathbf{s} \in A_i\}$, where commonly that value is the spatial integral or the spatial average over A_i . Define the discretised latent spatial process $Y(\cdot)$ evaluated over the N BAUs as $\mathbf{Y} \equiv (Y_1, \dots, Y_N)^\top$, where $Y_i \equiv Y(A_i)$, $i = 1, \dots, N$. Then, a vectorised version of (1) is

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\alpha} + \mathbf{S}\boldsymbol{\eta} + \boldsymbol{\xi}, \tag{2}$$

where \mathbf{T} and \mathbf{S} are known design matrices constructed from $\mathbf{t}(\cdot)$ and $\boldsymbol{\phi}(\cdot)$ respectively, $\boldsymbol{\alpha}$ is a fixed effect, and $\boldsymbol{\xi}$ is a vector associated with the fine-scale process which, like $\boldsymbol{\eta}$, is random.

As in **FRK** v1, the elements of $\boldsymbol{\xi}$ are often modelled as independent and identically distributed (i.i.d.) Gaussian random variables with mean zero and variance σ_ξ^2 , and $\boldsymbol{\eta}$ is modelled as a mean-zero multivariate-Gaussian random vector with covariance matrix $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$.

In **FRK** v2, $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$ is modelled either as \mathbf{K} or as \mathbf{Q}^{-1} , where \mathbf{Q} is a precision matrix. Both formulations use block-diagonal matrices, so that basis-function coefficients between basis-function resolutions are independent; see Appendix B for how the intra-resolution dependencies with \mathbf{K} and \mathbf{Q} are modelled. Although both \mathbf{K} and \mathbf{Q} are generally sparse, use of \mathbf{Q} instead of \mathbf{K} is typically computationally advantageous. Irrespective of the parameterisation, we assume that $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$ depends on an unknown parameter vector $\boldsymbol{\vartheta}$.

Following standard generalised-linear-model theory ([McCullagh and Nelder 1989](#)), we use an invertible link function, $g(\cdot)$, to model $Y(\cdot)$ as a transformation of a mean process, $\mu(\cdot)$:

$$g(\mu(\mathbf{s})) = Y(\mathbf{s}); \quad \mathbf{s} \in D. \tag{3}$$

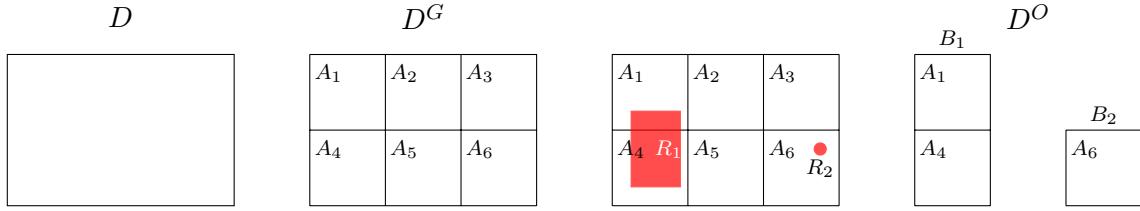


Figure 1: An illustration of how the spatial domain, D , is discretised into the set D^G of BAUs, and how the observation domain, D^O , is derived from the observation supports. (Left) The continuous spatial domain, D . (Centre left) The spatial domain discretised into $N = 6$ BAUs, $D^G \equiv \{A_i : i = 1, \dots, 6\}$. (Centre right) D^G superimposed with $m = 2$ observations, one that is areally-referenced (R_1), and one that is point-referenced (R_2). (Right) The observation domain, $D^O \equiv \{B_j : j = 1, 2\}$, where $B_1 \equiv A_1 \cup A_4$ and $B_2 \equiv A_6$.

Therefore, the mean process evaluated over the BAUs is $\boldsymbol{\mu} \equiv (\mu_i : i = 1, \dots, N)^\top$, where $\mu_i = g^{-1}(Y_i)$, $i = 1, \dots, N$, and $g^{-1}(\cdot)$ is the inverse link function. We sometimes write $\boldsymbol{\mu} = g^{-1}(\mathbf{Y})$ and $\mathbf{Y} = g(\boldsymbol{\mu})$, where the functions are applied element-wise.

2.2. The data layer

We denote the vector of m observations (the data vector) as $\mathbf{Z} \equiv (Z_1, \dots, Z_m)^\top$. Each datum is originally associated with a spatial support, $R_j \subseteq D$, $j = 1, \dots, m$, which we associate to one or more BAUs. In practice, these spatial supports may not coincide with entire BAUs and, when this is the case, in **FRK v2** we assume that a spatial support contains a BAU if and only if there is a non-empty intersection between the BAU and the spatial support. That is, we write the indices of the BAUs associated with spatial support R_j as $c_j \equiv \{i : A_i \cap R_j \neq \emptyset\}$, for $j = 1, \dots, m$. We then define the set of observation supports in terms of BAUs as $D^O \equiv \{B_j : j = 1, \dots, m\}$, where $B_j \equiv \bigcup_{i \in c_j} A_i$ is the package's representation of R_j in terms of BAUs. Figure 1 shows a pedagogical example with $m = 2$ observations illustrating the relationship between the continuous domain D , the BAUs $\{A_1, \dots, A_6\}$, the original spatial supports $\{R_1, R_2\}$, and the observation supports $\{B_1, B_2\}$.

Define the conditional mean of the data as $\boldsymbol{\mu}_Z \equiv (\mathbb{E}(Z_1 | \boldsymbol{\mu}), \dots, \mathbb{E}(Z_m | \boldsymbol{\mu}))^\top$, where henceforth we treat the supports of Z_1, \dots, Z_m as B_1, \dots, B_m , respectively. Since each $B_j \in D^O$ is either a BAU or a union of BAUs, one can construct an $m \times N$ matrix

$$\mathbf{C}_Z \equiv \left(w_{ij} \mathbb{I}(i \in c_j) : i = 1, \dots, N; j = 1, \dots, m \right), \quad (4)$$

where $\mathbb{I}(\cdot)$ is the indicator function, such that

$$\boldsymbol{\mu}_Z = \mathbf{C}_Z \boldsymbol{\mu}. \quad (5)$$

The matrix \mathbf{C}_Z aggregates the BAU-level process $\boldsymbol{\mu}$ over the observation supports and, depending on the weights in (4), it can correspond to a weighted average or a weighted sum over the BAUs. See Appendix C for details.

Denoting the j th element of $\boldsymbol{\mu}_Z$ by μ_{Z_j} , we assume that

$$[Z_j | \boldsymbol{\mu}, \psi] = \text{EF}(\mu_{Z_j}, \psi); \quad j = 1, \dots, m, \quad (6)$$

where EF corresponds to a probability distribution in the exponential family with dispersion parameter ψ and, for generic random quantities A and B , $[A \mid B]$ denotes the probability distribution of A given B . We assume that ψ is spatially invariant (note that ψ is equal to 1 for the binomial, negative-binomial, and Poisson distributions).

Together, (5) and (6) imply that a given observation depends only on the values of the mean process $\boldsymbol{\mu}$ at the corresponding observation support. Further, we assume that all observations are conditionally independent given the latent spatial process, and that they are all from the same exponential family member. Specifically,

$$[\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi] = \prod_{j=1}^m \text{EF}(\mu_{Zj}, \psi).$$

As we only consider data models in the exponential family, $\ln [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi]$ may be written as

$$\ln [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi] = \sum_{j=1}^m \left\{ \frac{Z_j \lambda(\mu_{Zj}) - b(\lambda(\mu_{Zj}))}{a(\psi)} + c(Z_j, \psi) \right\}, \quad (7)$$

where $a(\cdot)$, $b(\cdot)$, and $c(\cdot, \cdot)$ are deterministic functions specific to the chosen exponential family member, and $\lambda(\cdot)$ is the canonical parameter.

The model employed by **FRK** v2 can be summarised as follows.

$$Z_j \mid \boldsymbol{\mu}_Z, \psi \stackrel{\text{ind}}{\sim} \text{EF}(\mu_{Zj}, \psi); \quad j = 1, \dots, m, \quad (8)$$

$$\boldsymbol{\mu}_Z = \mathbf{C}_Z \boldsymbol{\mu}, \quad (9)$$

$$g(\boldsymbol{\mu}) = \mathbf{Y}, \quad (10)$$

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\alpha} + \mathbf{S}\boldsymbol{\eta} + \boldsymbol{\xi}, \quad (11)$$

$$\boldsymbol{\eta} \mid \boldsymbol{\vartheta} \sim \text{Gau}(\mathbf{0}, \mathbf{Q}^{-1}), \quad (12)$$

$$\boldsymbol{\xi} \mid \sigma_\xi^2 \sim \text{Gau}(\mathbf{0}, \sigma_\xi^2 \mathbf{V}), \quad (13)$$

where \mathbf{V} is a known, positive-definite diagonal matrix which, in the absence of problem specific fine-scale information, can simply be set to \mathbf{I} , and σ_ξ^2 is either unknown and estimated, or provided by the user. In a spatio-temporal setting, a more complex model for $\boldsymbol{\xi}$ is allowed; see Section 2.6. Note that **FRK** v2 is backwards compatible, since an identity link function and a Gaussian data model in (8) yields the model used in **FRK** v1.

2.3. Estimation

Noting that $\boldsymbol{\mu}_Z$ is, through (9)–(11), completely determined by $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, and $\boldsymbol{\xi}$, the complete-data likelihood function for our model is

$$L(\boldsymbol{\theta}; \mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}) \equiv [\mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi} \mid \boldsymbol{\theta}] = [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi][\boldsymbol{\eta} \mid \boldsymbol{\vartheta}][\boldsymbol{\xi} \mid \sigma_\xi^2], \quad (14)$$

where $\boldsymbol{\theta} \equiv (\boldsymbol{\alpha}^\top, \boldsymbol{\vartheta}^\top, \sigma_\xi^2, \psi)^\top$, and recall that $\boldsymbol{\vartheta}$ denotes the variance-covariance components associated with either \mathbf{K} or \mathbf{Q} . The complete-data log-likelihood function, $\ell(\boldsymbol{\theta}; \mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi})$, is simply the logarithm of (14). Under the modelling assumptions (8)–(13), the conditional density functions $[\boldsymbol{\eta} \mid \boldsymbol{\vartheta}]$ and $[\boldsymbol{\xi} \mid \sigma_\xi^2]$ are invariant to the specified link function and the assumed distribution of the response variable.

The observed-data likelihood, which depends on the observations \mathbf{Z} and not on the unobserved random effects $\mathbf{u} \equiv (\boldsymbol{\eta}^\top, \boldsymbol{\xi}^\top)^\top$, is given by integrating out \mathbf{u} from (14):

$$L^*(\boldsymbol{\theta}; \mathbf{Z}) \equiv \int_{\mathbb{R}^p} L(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u}) d\mathbf{u}, \quad (15)$$

where p is the total number of random effects in the model. The observed-data log-likelihood function is $\ell^*(\boldsymbol{\theta}; \mathbf{Z}) \equiv \log L^*(\boldsymbol{\theta}; \mathbf{Z})$. When the data are non-Gaussian, the integral in (15) is typically intractable and must be approximated. In **FRK** v2, a Laplace approximation is used, which we now briefly describe.

Let $\hat{\mathbf{u}} \equiv \hat{\mathbf{u}}(\boldsymbol{\theta}, \mathbf{Z})$ be a mode of $\ell(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u})$ with respect to \mathbf{u} , and let

$$\mathbf{H} \equiv -\left(\nabla_{\mathbf{u}} \nabla_{\mathbf{u}} \ell(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u})|_{\mathbf{u}=\hat{\mathbf{u}}}\right)^{-1},$$

where $\nabla_{\mathbf{u}}$ denotes the gradient with respect to \mathbf{u} . A second-order Taylor-series approximation of $\ell(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u})$ about $\mathbf{u} = \hat{\mathbf{u}}$ results in an approximation of (14) that has the form of an un-normalised Gaussian density in terms of \mathbf{u} , with mean vector $\hat{\mathbf{u}}$ and covariance matrix \mathbf{H} . Substitution of this approximation into (15) and evaluation of the integral, yields the Laplace approximation of the observed-data likelihood, $L^*(\boldsymbol{\theta}; \mathbf{Z}) \approx L(\boldsymbol{\theta}; \mathbf{Z}, \hat{\mathbf{u}})(2\pi)^{\frac{p}{2}} |\mathbf{H}|^{\frac{1}{2}}$. For a more detailed discussion on the Laplace approximation and its properties, see, for example, [Tierney and Kadane \(1986\)](#), [Rue and Martino \(2007\)](#), and [Rue et al. \(2009\)](#).

Note that $[\mathbf{u} | \mathbf{Z}, \boldsymbol{\theta}] \propto [\mathbf{Z}, \mathbf{u} | \boldsymbol{\theta}]$, which is equal to the complete-data likelihood function, $L(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u})$. Therefore, since the Laplace approximation replaces $L(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u})$ with a term that has the form of an un-normalised Gaussian density in terms of \mathbf{u} , it follows that, approximately, $\mathbf{u} | \mathbf{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\mathbf{u}}, \mathbf{H})$. In the software we use (see below), estimates of $\hat{\mathbf{u}}$ and \mathbf{H}^{-1} are provided, which makes prediction of \mathbf{u} and any function of it straightforward via the predictive distribution and its MC simulation (see Section 2.4).

Model fitting with **TMB**

FRK v2 supplies the R package **TMB** ([Kristensen, Nielsen, Berg, Skaug, and Bell 2016](#)) with a C++ template function that defines $\ell(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u})$. **TMB** then computes the Laplace approximation of the observed-data log-likelihood, $\ell^*(\boldsymbol{\theta}; \mathbf{Z})$, and it automatically computes its derivatives; these quantities are then invoked via a user-defined optimising function (`nlminb()` is used by default). **TMB** uses **CppAD** ([Bell 2005](#)) for automatic differentiation, and it uses the linear-algebra libraries **Eigen** ([Guennebaud, Jacob et al. 2010](#)) and **Matrix** ([Bates, Maechler, and Davis 2019](#)) for vector and matrix operations in C++ and R, respectively. Use of these packages yields high computational efficiency.

Note that all unknown quantities are treated as random in **TMB**. To retain **FRK** v1's mixed-model interpretation, we fix the model parameters and fixed effects to their posterior-mode estimates and then treat them as non-random quantities.

2.4. Prediction and uncertainty quantification

There are three principal quantities that could be of interest to the user, namely the latent process \mathbf{Y} and mean process $\boldsymbol{\mu}$ in (10), and data at unobserved locations. Recall that the Laplace approximation approximates the conditional distribution of $\mathbf{u} \equiv (\boldsymbol{\eta}^\top, \boldsymbol{\xi}^\top)^\top$ as $\mathbf{u} | \mathbf{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\mathbf{u}}, \mathbf{H})$; since \mathbf{Y} is a linear function of \mathbf{u} , approximate inference on \mathbf{Y} can be

carried out using well-known formulas. However, the posterior distribution of a non-linear function of \mathbf{Y} , for example the mean $\boldsymbol{\mu}$ in (10), is typically not available in closed form, and some approximation is required. In **FRK** v2 we therefore use a Monte Carlo (MC) approach to inference on non-linear functions of \mathbf{Y} , by first drawing a sample from the approximate conditional distribution of \mathbf{u} and then transforming the sample accordingly.

Recall that $\mathbf{Y} = \mathbf{T}\boldsymbol{\alpha} + \mathbf{S}\boldsymbol{\eta} + \boldsymbol{\xi}$, which can be rewritten as $\mathbf{Y} = \mathbf{T}\boldsymbol{\alpha} + [\mathbf{S} \ \mathbf{I}] \mathbf{u}$. We thus define \mathbf{Y}_{MC} , an $N \times n_{\text{MC}}$ matrix whose columns are the n_{MC} MC samples from $[\mathbf{Y} \mid \mathbf{Z}, \boldsymbol{\theta}]$, as

$$\mathbf{Y}_{\text{MC}} \equiv \mathbf{T}\hat{\mathbf{A}} + [\mathbf{S} \ \mathbf{I}] \mathbf{U}, \quad (16)$$

where each of the n_{MC} columns of the matrix $\hat{\mathbf{A}}$ is the estimate of $\boldsymbol{\alpha}$, and each of the n_{MC} columns of the matrix \mathbf{U} is a draw from $\mathbf{u} \mid \mathbf{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\mathbf{u}}, \mathbf{H})$. We obtain MC samples of the N -dimensional vector $\boldsymbol{\mu}$ from $[\boldsymbol{\mu} \mid \mathbf{Z}, \boldsymbol{\theta}]$ via the $N \times n_{\text{MC}}$ matrix $\mathbf{M} \equiv g^{-1}(\mathbf{Y}_{\text{MC}})$, where $g^{-1}(\cdot)$ is applied element-wise.

FRK v2 also allows prediction of data over all N BAUs, which we write as $\mathbf{Z}^* \equiv (Z_1^*, \dots, Z_N^*)^\top$. We assume that these data are from the same exponential family model as that of the original data, \mathbf{Z} . MC samples of \mathbf{Z}^* can then be constructed straightforwardly using \mathbf{M} . Note that **FRK** v2 provides the user with \mathbf{M} which, if needed, could be used to predict data with link function $g(\cdot)$ but from a distribution that is different from that of the original data.

For each quantity, we use the posterior expectation as the predictor, which can be estimated by simply taking row-wise averages of the matrices of samples defined above. In a Gaussian setting, a commonly used metric for uncertainty quantification is the root-mean-squared prediction error (RMSPE). In a non-Gaussian setting, it can be difficult to interpret the RMSPE, and it is often more intuitive to quantify uncertainty through the width of the prediction intervals. Hence, in **FRK** v2, we also use the MC sampling approach described above to compute user-specified percentiles of the predictive distribution.

Arbitrary prediction regions

Often, one does not wish to predict over single BAUs but over regions spanning multiple BAUs, $\{\tilde{R}_l, l = 1, \dots, N_P\}$, where N_P is the number of prediction regions. These regions may overlap and may not coincide with entire BAUs: Our criterion for determining whether a prediction region contains a particular BAU is the same as that used for the spatial supports originally associated with the observations (see Section 2.2). That is, we write the indices of the BAUs associated with \tilde{R}_l as $\tilde{c}_l \equiv \{i : A_i \cap \tilde{R}_l \neq \emptyset\}$, for $l = 1, \dots, N_P$. We then define the set of prediction regions in terms of BAUs as $D^P \equiv \{\tilde{B}_l : l = 1, \dots, N_P\}$, where $\tilde{B}_l \equiv \cup_{i \in \tilde{c}_l} A_i$ is the package's representation of \tilde{R}_l in terms of BAUs.

Prediction of $\boldsymbol{\mu}_P \equiv (\mu_{P,1}, \dots, \mu_{P,N_P})^\top$ over D^P requires some form of aggregation across the associated BAUs. In **FRK** v2, we aggregate the mean process $\boldsymbol{\mu}$ over the associated BAUs. We stress that this is different from aggregation of data (which would lead to a different model for dealing with change-of-support). Just as $\boldsymbol{\mu}_Z$ was constructed from the BAU-level mean process $\boldsymbol{\mu}$ via the matrix \mathbf{C}_Z given by (4), since each \tilde{B}_l is a BAU or a union of BAUs, one can construct an $N_P \times N$ matrix

$$\mathbf{C}_P \equiv (\tilde{w}_{il} \mathbb{I}(i \in \tilde{c}_l) : i = 1, \dots, N; l = 1, \dots, N_P), \quad (17)$$

such that

$$\boldsymbol{\mu}_P = \mathbf{C}_P \boldsymbol{\mu}. \quad (18)$$

For consistency between the model fitting and prediction stages, **FRK** v2 enforces \mathbf{C}_P to have the same qualitative behaviour as \mathbf{C}_Z (i.e., if \mathbf{C}_Z corresponds to a weighted average, then so too does \mathbf{C}_P). See Appendix C for details.

MC samples of $\boldsymbol{\mu}_P \mid \mathbf{Z}, \boldsymbol{\theta}$ are constructed via $\mathbf{M}_P \equiv \mathbf{C}_P \mathbf{M}$, where recall that the columns of \mathbf{M} consist of MC samples from $[\boldsymbol{\mu} \mid \mathbf{Z}, \boldsymbol{\theta}]$. Predictions and uncertainty quantification of the predictions can then be computed straightforwardly from \mathbf{M}_P . **FRK** v2 also allows inference on data $\{Z_{P,1}^*, \dots, Z_{P,N_P}^*\}$ over aggregations of BAUs, $\{\tilde{B}_1, \dots, \tilde{B}_{N_P}\}$. We assume that these data are from the same exponential family model as that of the original data, \mathbf{Z} . MC samples of $\mathbf{Z}_P^* \equiv (Z_{P,1}^*, \dots, Z_{P,N_P}^*)^\top$ can then be constructed straightforwardly using \mathbf{M}_P . Note that **FRK** v2 provides the user with \mathbf{M}_P which, if needed, can be used to predict data with link function $g(\cdot)$ but from a distribution that is different from that of the original data.

2.5. Distributions with size parameters

Two distributions considered in this framework, namely the binomial distribution and the negative-binomial distribution, have an assumed-known ‘size’ parameter and a ‘probability of success’ parameter. Given the vector of size parameters associated with the data, $\mathbf{k}_Z \equiv (k_{Z_1}, \dots, k_{Z_m})^\top$, the parameterisation used in **FRK** v2 assumes that Z_j represents either the number of ‘successes’ from k_{Z_j} trials (binomial data model) or that it represents the number of failures before k_{Z_j} successes (negative-binomial data model). Some sources use different parameterisations for the negative-binomial distribution: The parameterisation used in **FRK** v2 is the same as that used in the R package **stats** (R Core Team 2021).

Software that cater for these distributions typically allow ‘link’ functions such as the logit, probit, and complementary log-log functions. In **FRK** v2, these functions are available to link the latent spatial process, $Y(\cdot)$, to a probability process, $\pi(\cdot)$:

$$f(\pi(\mathbf{s})) = Y(\mathbf{s}), \quad \mathbf{s} \in D,$$

where $f(\cdot)$ is one of the aforementioned functions whose inverse has a range of $(0, 1)$. Therefore, the probability process evaluated over the BAUs is $\boldsymbol{\pi} \equiv (\pi_i : i = 1, \dots, N)^\top$, where

$$\pi_i = f^{-1}(Y_i), \quad i = 1, \dots, N. \tag{19}$$

Next, we link the BAU-level mean process to the BAU-level probability process,

$$h(\mu_i; k_i) = \pi_i, \quad i = 1, \dots, N, \tag{20}$$

where $h(\cdot; \cdot)$ is derived from the expectation of the response distribution (see Appendix D for details), and $\mathbf{k} \equiv (k_1, \dots, k_N)^\top$ is the vector of BAU-level size parameters. For example, a binomial data model results in $\mu_i/k_i = \pi_i$, for $i = 1, \dots, N$. When modelling negative-binomial data, other popular link functions include the log and square-root functions. In these cases, we use the model $\mu_i = k_i g^{-1}(Y_i)$, $i = 1, \dots, N$, and the elements of $\boldsymbol{\pi}$ are obtained using (20). Irrespective of the chosen link function, we define the probability process over the observation supports in D^O as $\boldsymbol{\pi}_Z \equiv (\pi_{Z_j} : j = 1, \dots, m)^\top$, where

$$\pi_{Z_j} = h(\mu_{Z_j}; k_{Z_j}), \quad j = 1, \dots, m. \tag{21}$$

When model fitting, the BAU-level size parameters \mathbf{k} are needed to compute the BAU-level mean process in (20). The user must supply these size parameters either through the data or

though the BAUs. How this is done depends on whether the data are areal or point-referenced, and whether they overlap common BAUs or not; details are provided in the package manual.

Now, define the prediction-region size parameters as $\mathbf{k}_P \equiv (\sum_{i \in \tilde{c}_l} k_i : l = 1, \dots, N_P)^\top$, where recall that \tilde{c}_l denotes the indices of the BAUs associated with the prediction region $\tilde{B}_l \in D^P$. Then the probability process evaluated over D^P is

$$\boldsymbol{\pi}_P \equiv h(\boldsymbol{\mu}_P; \mathbf{k}_P), \quad (22)$$

where $h(\cdot; \cdot)$ is applied element-wise. When predicting, BAU-level size parameters are needed to compute the predictive distribution of $\boldsymbol{\mu}$, $\boldsymbol{\pi}_P$, and $\boldsymbol{\mu}_P$. If these size parameters are not available at unobserved BAUs, the user can still obtain predictions of $\boldsymbol{\pi}$ if the user chooses to use a logit, probit, or complementary log-log link function, since then $\boldsymbol{\pi}$ is linked directly to \mathbf{Y} in (19).

In most applications that consider binomial or negative-binomial data models, the conditional mean of an observation is treated as a simple aggregate of the underlying mean process. Therefore, with these distributions, **FRK** v2 enforces the matrices \mathbf{C}_Z and \mathbf{C}_P in (4) and (17), respectively, to correspond to a simple, unweighted summation over the BAUs. See Appendix C for details.

2.6. Spatio-temporal framework

FRK v2 accommodates spatio-temporal data by using spatio-temporal basis functions constructed via a tensor product of spatial and temporal basis functions. Since one often requires several thousand basis functions in a spatio-temporal setting, we focus here on the case where $\boldsymbol{\eta}$ is modelled using the sparse precision matrix \mathbf{Q} .

Let r_t and r_s denote the number of temporal and spatial basis functions, respectively. Denote \mathbf{Q}_t and \mathbf{Q}_s as the precision matrices of the random coefficients associated with the temporal basis functions and the spatial basis functions, respectively. We model the $r_t r_s \times r_t r_s$ precision matrix of the random coefficients associated with the $r_t r_s$ spatio-temporal basis functions as $\mathbf{Q} = \mathbf{Q}_t \otimes \mathbf{Q}_s$, where \otimes is the Kronecker product. This form for \mathbf{Q} leads to significant computational savings. For the random coefficients associated with the temporal basis functions, **FRK** v2 uses a first-order autoregressive model.

Recall from the spatial-only case that **FRK** v2 assumes $\boldsymbol{\xi} \sim \text{Gau}(\mathbf{0}, \sigma_\xi^2 \mathbf{V})$, where \mathbf{V} is a known, positive-definite diagonal matrix that can be set to \mathbf{I} in the absence of problem specific fine-scale information. In a spatio-temporal setting, it is possible that each spatial BAU is observed multiple times. In these situations, **FRK** v2 also allows each spatial BAU to be associated with its own fine-scale variance parameter. Specifically, let N_s and N_t denote the number of spatial and temporal BAUs, respectively, so that $N = N_s N_t$. Then, **FRK** v2 also allows the model $\boldsymbol{\xi} \sim \text{Gau}(\mathbf{0}, \boldsymbol{\Sigma}_\xi)$ where, assuming that the BAUs are ordered such that space runs faster than time,

$$\boldsymbol{\Sigma}_\xi \equiv \text{diag}(\underbrace{\text{vec}(\boldsymbol{\sigma}_\xi^2, \dots, \boldsymbol{\sigma}_\xi^2)}_{N_t \text{ times}}) \odot \mathbf{V}, \quad (23)$$

$\boldsymbol{\sigma}_\xi^2 \equiv (\sigma_{\xi,1}^2, \dots, \sigma_{\xi,N_s}^2)^\top$, $\text{vec}(\cdot)$ ‘stacks’ its arguments into a single vector, $\text{diag}(\cdot)$ returns a diagonal matrix from a vector argument, and \odot denotes element-wise multiplication. This model may be advantageous when the variance of the fine-scale component is spatially varying, when the number of spatial BAUs (and hence the number of fine-scale variance parameters

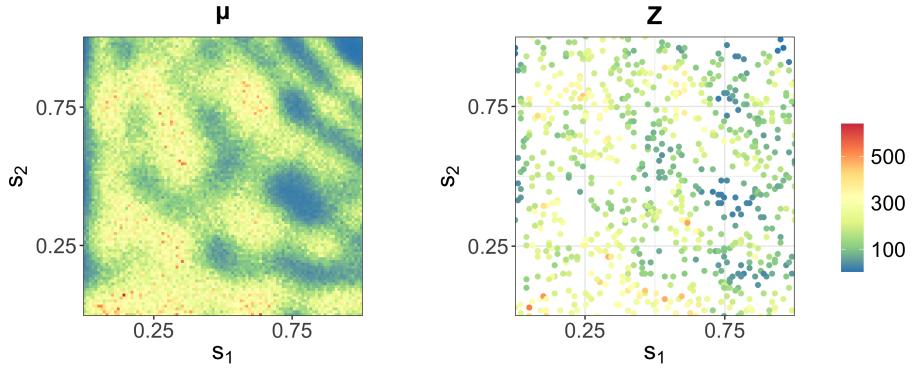


Figure 2: Simulated, point-referenced, Poisson data set used in the illustrative example of Section 3.1. (Left) True mean process evaluated over the BAUs. (Right) Simulated data.

to estimate) is relatively low, and when we have observations from each spatial BAU at many time-points; see, for instance, the example presented in Section 4.4.

3. New features and their usage

We now focus on the new features in **FRK** v2, an overview of which is presented in Table 1. In Section 3.1, we illustrate the use of **FRK** v2 with non-Gaussian spatial data. In Section 3.2, we demonstrate how the important tasks of model selection and model validation can be performed with **FRK** v2. In Section 3.3, we show the potential improvement in predictive performance of **FRK** v2 over **FRK** v1 when the data are Gaussian, owing to an increase in the maximum number of basis functions allowed in **FRK** v2. All results presented in the remainder of this paper can be generated using the reproducible code at https://github.com/msainsburydale/FRKv2_src.

3.1. Non-Gaussian data

The primary new feature in **FRK** v2 is the package’s ability to cater for non-Gaussian data models. The available exponential-family-members include the Gaussian, Poisson, gamma, inverse-Gaussian, negative-binomial, and binomial distributions, and these distributions can be used in combination with the identity, inverse, log, square-root, logit, probit, and complementary-log-log link functions.

For illustration, and so that readers can familiarise themselves with the workflow of **FRK** v2, we now analyse a simulated Poisson data set containing 750 observations at spatial locations shown in Figure 2. The true mean process evaluated over the BAUs, μ , is also shown in Figure 2: It was constructed by passing a sum of trigonometric functions through the exponential function. In what follows, $\text{Poi}(\mu)$ refers to a probability distribution on the non-negative integers with probability mass function (PMF) $\{e^{-\mu}\mu^z/z! : z = 0, 1, \dots\}$ for $\mu > 0$.

The first step when using **FRK** is to create basis functions and BAUs, which can be done automatically using the helper functions, `auto_BAUs()` and `auto_basis()`; see Appendix A for details. Next, an ‘SRE’ object is constructed using `SRE()`, within which we specify the data model, the link function, and the parameterisation of $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$. In this example, we model counts using a Poisson data model, $Z_j | \boldsymbol{\mu} \sim \text{Poi}(\mu_{Z_j})$, for $j = 1, \dots, m$, and we use the log

Function	Argument	Use
<code>SRE()</code> / <code>FRK()</code>	<code>response</code>	String indicating the response distribution.
	<code>link</code>	String indicating the link function.
	<code>K_type</code>	String indicating the parameterisation of $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$; the newly permissible value, "precision", indicates that a sparse precision matrix should be used.
	<code>normalise_wts</code>	Flag controlling whether the weights in \mathbf{C}_Z and \mathbf{C}_P should be normalised or not.
	<code>fs_by_spatial_BAU</code>	Flag controlling whether each spatial BAU is given its own fine-scale variance parameter; only applicable in a spatio-temporal setting.
<code>SRE.fit()</code> / <code>FRK()</code>	<code>method</code>	String indicating the method of model fitting: "TMB" is required whenever a non-Gaussian data model or non-identity link function is used.
	<code>known_sigma2fs</code>	Positive number at which to fix the fine-scale variance.
<code>predict()</code>	<code>type</code>	Vector of strings indicating the quantities of interest for which inference is made. The inclusion of "link" indicates that inference on the latent process (\mathbf{Y}) is made; the inclusion of "mean" indicates that inference on the mean process ($\boldsymbol{\mu}$ or $\boldsymbol{\mu}_P$) and, if applicable, the probability process ($\boldsymbol{\pi}$ or $\boldsymbol{\pi}_P$) is made; and the inclusion of "response" indicates that inference on the data (\mathbf{Z}^* or \mathbf{Z}_P^*) is made.
	<code>percentiles</code>	Numeric vector indicating the percentiles of the predictive distribution(s) to be returned.
	<code>nsim</code>	The number of MC samples at each BAU.
<code>auto_BAUs()</code>	<code>spatial_BAUs</code>	The spatial BAUs in a spatio-temporal setting (constructed automatically from the data by default).
<code>plot()</code>	-	Visualise the data, predictions, and uncertainty quantification of the predictions given an 'SRE' object and the object resulting from a call to <code>predict()</code> .

Table 1: Important new or augmented function arguments in **FRK** v2. The package now also provides methods for `AIC()`, `BIC()`, `fitted()`, `residuals()`, and `simulate()`.

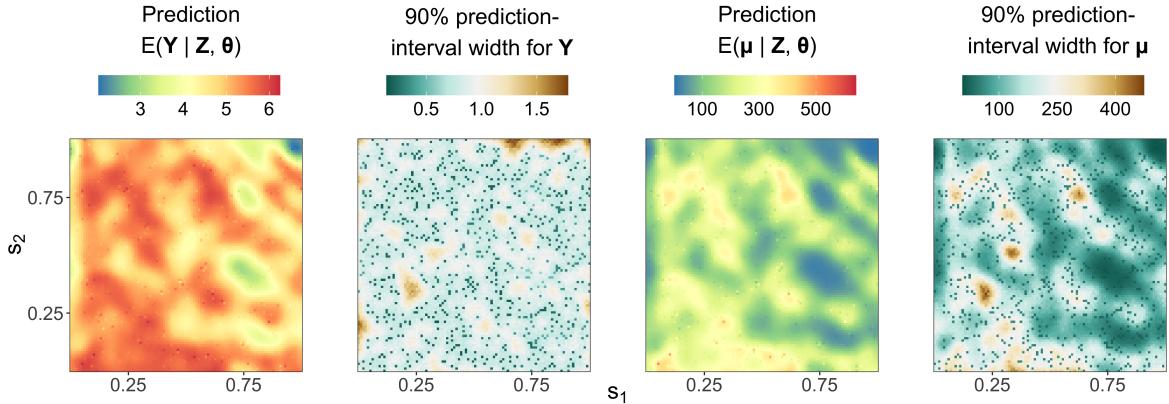


Figure 3: Predictions and prediction-interval widths returned by **FRK** v2 when applied to the simulated Poisson data shown in the right panel of Figure 2. (Left) Prediction of \mathbf{Y} , the latent process over the BAUs. (Centre left) Width of the 90% prediction interval for each element of \mathbf{Y} . (Centre right) Prediction of $\boldsymbol{\mu} = g^{-1}(\mathbf{Y})$, the mean process over the BAUs. (Right) Width of the 90% prediction interval for each element of $\boldsymbol{\mu}$.

link function, $g(\cdot) = \log(\cdot)$. These choices are made by setting `response = "poisson"` and `link = "log"`. We then fit the model using `SRE.fit()`. These steps may also be performed with the convenient wrapper function `FRK()`. Note that when the data are non-Gaussian or when a non-identity link function is chosen, `FRK()` automatically enforces `method = "TMB"` and selects `K_type = "precision"`, which means that **TMB** is used for model fitting and that the basis-function coefficients are modelled via a sparse precision matrix, \mathbf{Q} :

```
R> S <- FRK(f = Z ~ 1, data = Poisson_data, response = "poisson", link = "log")
```

Prediction is done using `predict()`. The argument `type` specifies the quantities of interest for which predictions and uncertainty quantification of the predictions are desired. In this example, we set `type = c("link", "mean")` to obtain predictions for the latent spatial process, \mathbf{Y} , and the mean process, $\boldsymbol{\mu}$. The `percentiles` argument allows the computation of percentiles of the predictive distributions and hence computation of prediction intervals; if left unspecified, the 5th and 95th percentiles are returned:

```
R> pred <- predict(S, type = c("link", "mean"))
```

When `method = "TMB"`, the returned object from `predict()` is a ‘list’ containing two elements. The first element is an object of the same class as `newdata` (if `newdata` is unspecified, prediction is done over the BAUs) and contains the predictions and uncertainty quantification of the predictions for each term in `type`. The second element is a ‘list’ of matrices containing MC samples for each term in `type` at each prediction location.

Finally, a ‘list’ of ‘ggplot’ (Wickham 2016) objects of the predictions and their associated uncertainty can be generated using `plot()`:

```
R> plots <- plot(S, pred)
```

The ‘ggplot’ objects can be arranged easily on a grid using various dedicated packages. Figure 3 shows predictions and prediction-interval widths for the latent process evaluated over the BAUs, \mathbf{Y} , and the mean process evaluated over the BAUs, $\boldsymbol{\mu}$. The predictions of $\boldsymbol{\mu}$ are reasonable given the data and the true values shown in Figure 2. The prediction-interval widths for \mathbf{Y} overall, do not vary much, but they are larger in regions of data paucity and along the boundary of the spatial domain; on the other hand, the prediction-interval width for $\boldsymbol{\mu}$ is large when $\boldsymbol{\mu}$ is large, as can be expected when the response is Poisson distributed. The ‘bullseye’ points of low uncertainty, visible for both processes, correspond to the data locations. The point-like nature of this reduction in uncertainty arises from the fine-scale random process, $\xi(\cdot)$, being modelled as mutually independent at the BAU level: The fine-scale random effects at unobserved BAUs do not “borrow strength” from the inferred fine-scale random effect at neighbouring observed BAUs.

3.2. Model validation and selection

When working with (spatial) GLMMs, standard residuals are difficult to interpret because their expected properties (e.g., the expected dispersion) change with the fitted value. Due to these challenges, it is often easier to validate the model via simulation from the fitted model (Box 1980; Rubin 1984; Gelman, Meng, and Stern 1996; Gelman and Hill 2007, Ch. 24). This simulation-based approach is facilitated in **FRK** v2 with the function `simulate()`. In particular, simulations generated with `simulate()` may be used with the R package **DHARMA** (Hartig 2022) which, given observed data and simulations from a fitted model, computes interpretable, simulation-based quantile residuals (Cox and Snell 1968; Dunn and Smyth 1996). Under the true model, these residuals always follow a standard uniform distribution, which greatly facilitates their interpretation. Using the Poisson example of Section 3.1, one may create a ‘DHARMA’ object as follows:

```
R> DHARMA_object <- createDHARMA(
+   simulatedResponse = simulate(S),
+   observedResponse = Poisson_data$Z,
+   integerResponse = TRUE)
```

The ‘DHARMA’ object can then be used to construct residual plots and to perform a range of hypothesis tests, including tests for residual spatial dependence and zero-inflation, as well as arbitrary, user-defined tests. Figure 4 shows the result of calling `plot()` on the above ‘DHARMA’ object, which generates a QQ plot and a plot of the residuals against the predicted values. Although simulation-based residuals are a natural choice for the spatial GLMM employed by **FRK** v2, one may also compute the response or Pearson residuals using `residuals()`.

The tools used for model validation may also be employed in model selection. However, model selection also often involves an information criterion, which can typically be decomposed into a term that assesses goodness-of-fit and a term that penalises model complexity. Standard likelihood-based criteria are the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). The presence of dependent random effects (e.g., basis-function coefficients) complicates the interpretation of these criteria, since the effective number of parameters is more than the number of fixed effects and covariance parameters, and less than this number plus the number of dependent random effects. However, they are still often used in such settings and, hence, **FRK** v2 also provides methods for the functions `AIC()` and `BIC()`.

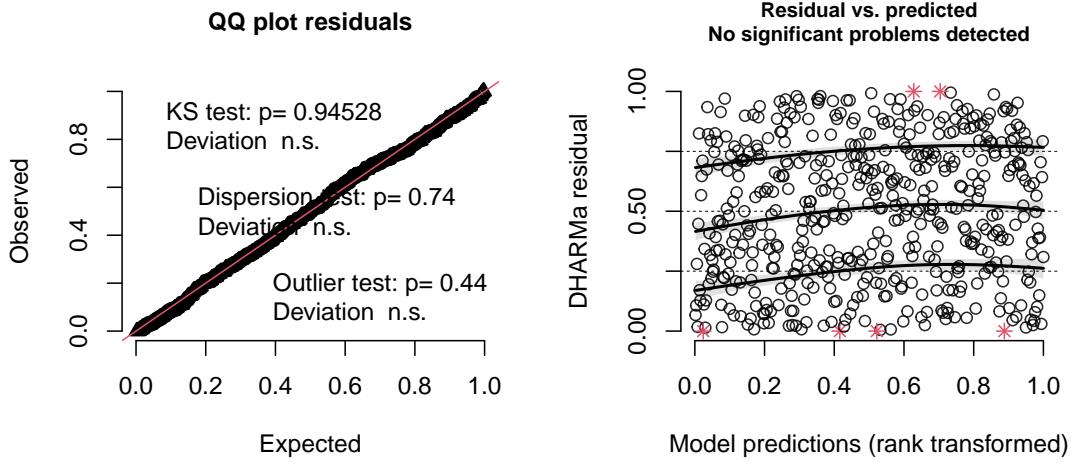


Figure 4: QQ plot (left) and the residuals against the predicted values (right) for the Poisson example of Section 3.1.

To demonstrate these approaches to model validation and selection, and to demonstrate that the model employed by **FRK** has appealing robustness properties with respect to the number of basis functions, we now repeat the analysis of Section 3.1 with models that employ one, two, three, and four resolutions of basis functions, respectively. Table 2 shows the results for each model. Clearly, predictive performance improves as the number of resolutions, and hence the number of basis functions, increases. However, the coverage remains accurate in all runs, implying that the model is able to accurately quantify uncertainty irrespective of the number of basis functions employed. This important property is due largely to the presence of the fine-scale random variation term, $\xi(\cdot)$, in (1). Further, the relative smoothness in the data (recall the right panel of Figure 2) suggests that only a moderate number of basis functions are needed for this application. Indeed, Table 2 shows that adding a fourth resolution of basis functions does not substantially improve the predictive performance, which is reflected by an increase in the AIC. Importantly, the use of more basis functions than is strictly necessary does not reduce predictive performance.

3.3. Increased numbers of basis functions

The efficiency of **TMB** and our use of precision matrix \mathbf{Q} instead of covariance matrix \mathbf{K} means that **FRK** v2 is well equipped to use a large number of basis functions. This is important, as the predictive performance of fixed rank kriging is often determined by the number of basis functions, as shown in the following experiment.

We re-ran the analysis for the comparative study of spatial-prediction methods published in Heaton, Datta, Finley, Furrer, Guinness, Guhaniyogi, Gerber, Gramacy, Hammerling, Katz-fuss, Lindgren, Nychka, Sun, and Zammit-Mangion (2019), which included spatial predictions from **FRK** v1. The data used in that study comprised a training data set and a test data set consisting of 105,569 observations and 42,740 observations, respectively. Table 3 replicates

Resolutions (basis functions)	RMSPE	CRPS	Cvg90	AIC
1 (9)	84.71	47.58	0.901	8894.9
2 (90)	58.60	30.97	0.878	8340.3
3 (819)	50.12	25.77	0.885	8117.3
4 (7380)	50.10	25.28	0.886	8121.4

Table 2: Diagnostics comparing the predictive performance when using a range of basis-function resolutions with point-referenced count data. The diagnostics are the root-mean-squared prediction error (RMPSE), the continuous ranked probability score (CRPS), and the empirical coverage (Cvg90) resulting from a prediction interval with a nominal coverage of 90% (see Appendix E for further details). The diagnostics are with regard to prediction of μ and they are averaged over all unobserved BAUs. The Akaike information criterion (AIC) for each model is also shown, and this is computed with respect to the observed data.

Method	MAE	RMSPE	CRPS	IS95	Cvg95	Run time (min.)	Cores
FRK v1	1.96	2.44	1.44	14.08	0.79	2.32	1
FRK v2	1.37	1.81	0.98	8.97	0.89	22.41	8
Gapfill	1.33	1.86	1.17	34.78	0.36	1.39	40
Lattice Krig	1.22	1.68	0.87	7.55	0.96	27.92	1
LAGP	1.65	2.08	1.17	10.81	0.83	2.27	40
Metakriging	2.08	2.50	1.44	10.77	0.89	2888.52	30
MRA	1.33	1.85	0.94	8.00	0.92	15.61	1
NNGP Conjugate	1.21	1.64	0.85	7.57	0.95	2.06	10
NNGP Response	1.24	1.68	0.87	7.50	0.94	42.85	10
Partition	1.41	1.80	1.02	10.49	0.86	79.98	55
Pred. Proc.	2.05	2.52	1.85	26.24	0.75	640.48	1
SPDE	1.10	1.53	0.83	8.85	0.97	120.33	2
Tapering	1.87	2.45	1.32	10.31	0.93	133.26	1
Periodic Embedding	1.29	1.79	0.91	7.44	0.93	9.81	1

Table 3: Scores for each method in the comparative study presented in Heaton *et al.* (2019). The scores are the mean absolute error (MAE), the root-mean-squared prediction error (RM-SPE), and the continuous ranked probability score (CRPS), and the empirical coverage (Cvg95) and the interval score (IS95) resulting from a prediction interval with a nominal coverage of 95% (see Appendix E for further details). Note that **FRK v2** was implemented in a different computing environment than the other models, and so its run time is not directly comparable. **FRK v2** was implemented using a machine with 16 GB of RAM and an Intel i7-9700 3.00GHz CPU with 8 cores. The other models were implemented using the powerful Becker computing environment with 256 GB of RAM and 2 Intel Xeon E5-2680 v4 2.40GHz CPUs with 14 cores each and 2 threads per core (Heaton *et al.* 2019).

Table 3 of Heaton *et al.* (2019), with an additional entry corresponding to **FRK v2**, wherein many more basis functions (12,144) were used than was practical with **FRK v1** (485). The results show that the increased number of basis functions significantly improved the diagnostic scores. To achieve these improvements, we only had to specify `nres = 4` rather than `nres = 3`; the rest of the code that was used in the comparative study was left unchanged.

4. Application and comparison studies

We now provide several application and comparison studies using **FRK** v2. In Section 4.1, we present a comparison study between **FRK** v2 and other packages that cater for non-Gaussian data models. In Section 4.2, we demonstrate block prediction using contaminated soil data. In Section 4.3, we use data on poverty figures in Sydney, Australia, to demonstrate the spatial change-of-support functionality of **FRK** v2 in a non-Gaussian setting; this is a useful example for readers wishing to ‘try out’ **FRK** v2 on real-world data. In Section 4.4, we provide a non-Gaussian spatio-temporal example through modelling crime counts in Chicago during the first two decades of the 21st century. For the use of **FRK** on the sphere, see [Zammit-Mangion and Cressie \(2021, Sec. 4.2\)](#) and the examples given in the package vignette.

4.1. Comparative study: MODIS cloud data

In this section, we compare out-of-sample predictions from **FRK** version 2.0.1 to those from the R packages **INLA** version 20.03.17 ([Lindgren and Rue 2015](#)), **spNNGP** version 0.1.4 ([Finley et al. 2020](#)), **spBayes** version 0.4.3 ([Finley et al. 2015](#)), and **mcmc** version 1.8.33 ([Wood 2017](#)) from a spatial binary data set. The data form an image of a cloud taken by the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument aboard the Aqua satellite ([MODIS Characterization Support Team 2015](#)) on 05 December 2018 01:00 UTC, over the South Pacific, just east of New Zealand. For this comparative study, we pre-processed the data as follows. First, we coarsened the image from over 10 million pixels to a more manageable 33,750 pixels, by creating a 150×225 grid and computing the average value of the response within each grid cell. Second, since the data provided by the MODIS instrument is continuous (namely spectral radiances in units of $\text{W/m}^2/\mu\text{m}/\text{st}$), we applied a threshold to obtain a binary version of the data. Specifically, pixels with spectral radiance greater than 7,000 $\text{W/m}^2/\mu\text{m}/\text{st}$ were labelled ‘Cloud’, and the remaining pixels were labelled ‘No-cloud’. This threshold was chosen to illustrate our methodology such that the resulting data have roughly equal proportions of ‘Cloud’ and ‘No-cloud’.

We considered two types of sampling schemes for model comparison. The first was missing-at-random (MR), where we randomly selected a sub-sample of pixels to act as training data. Under the MR sampling scheme, we randomly sampled 6,000 pixels for training, leaving 27,750 pixels for testing. The second sampling scheme, which we refer to as ‘missing-in-a-block’ (MB), involved using all pixels outside a central block for training, and using pixels inside the block for testing. The chosen testing block was a 30×30 square (900 pixels) in the middle of the spatial domain of interest, which meant that 32,850 pixels were used for training. The training and test sets under the two sampling schemes are shown in Figure 5. In what follows, $\text{Bin}(k, \pi)$ refers to a probability distribution on the non-negative integers with PMF $\{ \binom{k}{z} \pi^z (1 - \pi)^{k-z} : z = 0, \dots, k \}$ for $\pi \in (0, 1)$ and $k \in \{0, 1, \dots\}$. For the special case of $k = 1$, $\text{Bin}(1, \pi)$ is the Bernoulli distribution.

In this case study, the data model is $Z_j | \boldsymbol{\mu} \sim \text{Bin}(1, \pi_{Z_j})$, $j = 1, \dots, m$, where π_{Z_j} represents the probability of ‘Cloud’ over the pixel R_j . The software packages used in this study that allow use of this Bernoulli data model each required several modelling decisions, which had to be made in a way that balanced predictive performance and run time. We took a systematic approach to model-selection by splitting the training data set equally in two, and then using one half for model fitting and the other half for model evaluation. In this way, we were able to evaluate a large number of arguments for each package and choose the best combination

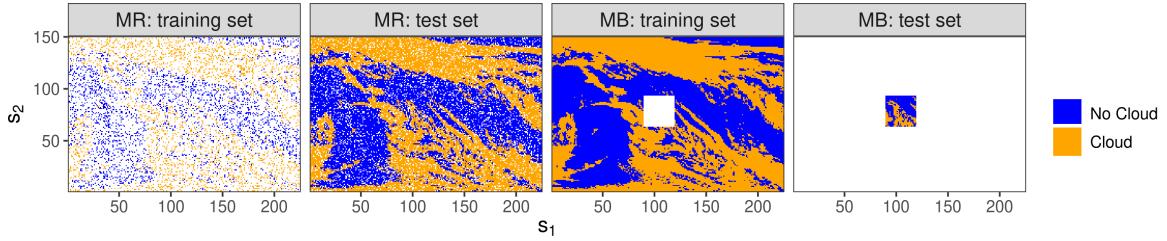


Figure 5: MODIS data used in the comparative study of Section 4.1; a white background is used to denote removed data. (Left) The missing-at-random (MR) sample used for training. (Centre left) The MR test data. (Centre right) The ‘missing-in-a-block’ (MB) sample used for training. (Right) The MB test data.

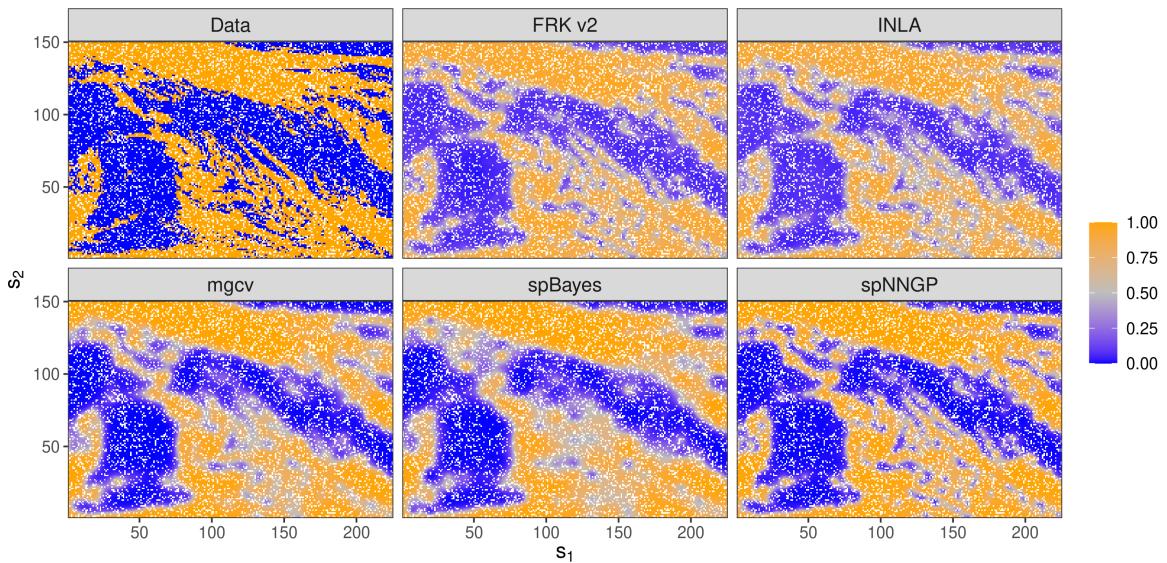


Figure 6: Predictions of the probability of ‘Cloud’ on the test set of the missing-at-random (MR) experiment shown in Figure 5. The corresponding test data is shown in the top-left panel. Note that the training locations are indicated by white pixels.

in terms of predictive performance and run time. For the methods requiring specification of a link function, we used the standard logit link function, $f(\pi) = \log(\frac{\pi}{1-\pi})$.

For **FRK** v2, we used four resolutions of basis functions, giving a total of 11,130 basis functions. For **INLA**, we discretised the domain into 13,494 elements. For **mgcv**, we used the `bam()` function, which is similar to the generalised-additive-model function `gam()` but optimised for large data sets, with 2,250 knots. For **spBayes**, we used 400 knots; increasing the number of knots further was computationally prohibitive (**spBayes** uses basis functions that depend on covariance-function parameters, so that computationally it can only handle a small number of knots). When using **spNNGP**, we found that the default option of considering 15 neighbours at a time was appropriate. For **spNNGP** and **spBayes**, we used 10,000 MCMC samples in total, with a burn-in of 6,000 samples and a thinning factor of 10. The number of cores used for **spNNGP** can be controlled through the argument `n.omp.threads`; choosing a value greater than 1 returned an error (a known issue documented in the **spNNGP** package manual) and, hence, our reported run-times for **spNNGP** are for a single core.

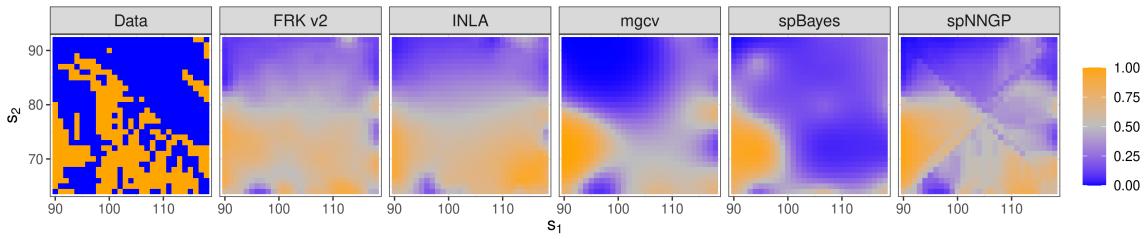


Figure 7: Predictions of the probability of ‘Cloud’ on the test set of the ‘missing-in-a-block’ (MB) experiment shown in Figure 5. Here, we only show the test locations, which correspond to the 30×30 block near the centre of the spatial domain; the test data in this block is shown in the left-most panel of this figure.

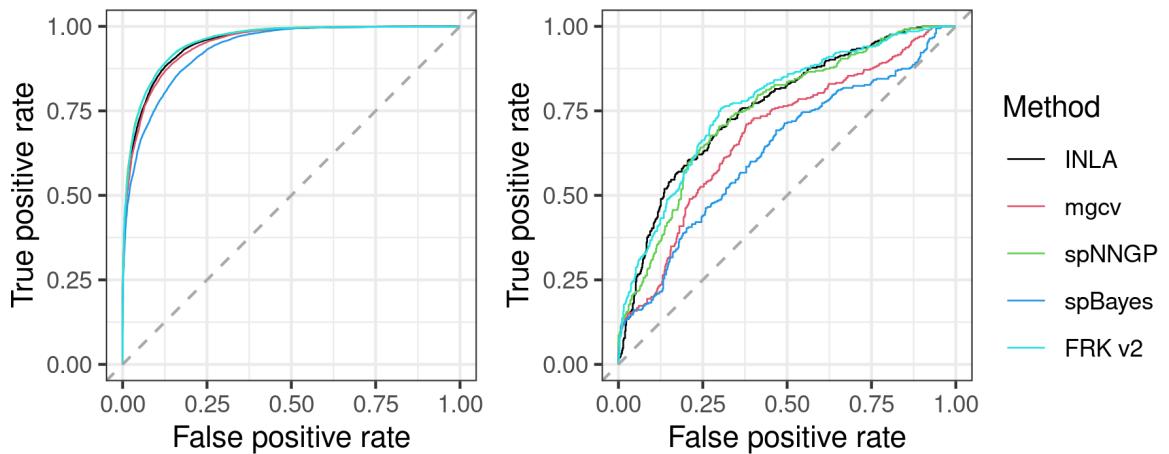


Figure 8: ROC curves for the training/test sets displayed in Figure 5. ROC curves generated in the ‘missing-at-random’ experiment (left) and the ‘missing-in-a-block’ experiment (right).

For each method and each sampling scheme, we predicted the probability of ‘Cloud’ at each pixel. Figure 6 shows the predictions resulting from the MR sampling scheme shown in Figure 5. The predictions from **FRK v2**, **INLA**, and **spNNGP** are similar, while the predictions from **mgcv** are slightly smoother than those from the aforementioned packages. The predictions of **spBayes** are even smoother, and this is due to the small number of knots it employs. Figure 7 shows the predictions resulting from the MB sampling scheme shown in Figure 5. Close inspection of the corresponding training data, shown in the centre right panel of Figure 5, indicates that predictions within the missing block are largely driven by observations immediately surrounding the block, as expected. The packages **FRK v2** and **INLA** return predictive probabilities close to 0.5, while **mgcv** and **spBayes** are more confident in their predictions. There is an interesting pattern in the **spNNGP** predictions; this is an expected artefact of the nearest-neighbour approach.

To assess predictive accuracy, we compared the predictions from all models using the Brier score (see Appendix E), and the area under the receiver operating characteristic (ROC) curve (AUC) (see, e.g., [Hastie, Tibshirani, and Friedman 2009](#), pg. 317). The Brier score assesses how close the predicted probability of ‘Cloud’ is to the truth; lower scores indicate more accurate predictions of the probability of ‘Cloud’. In contrast, higher AUC scores are

Scheme	Method	Brier score	AUC	Run time (min.)
MR	FRK v2	0.09	0.96	9.78
	INLA	0.09	0.95	6.48
	mcmc	0.09	0.95	26.53
	spBayes	0.11	0.93	73.01
	spNNGP	0.08	0.96	12.35
MB	FRK v2	0.19	0.77	31.74
	INLA	0.20	0.76	12.19
	mcmc	0.23	0.69	125.67
	spBayes	0.25	0.63	504.41
	spNNGP	0.20	0.75	65.30

Table 4: Diagnostic results for the MODIS comparison study.

preferred. The results for each method and each sampling scheme are reported in Table 4, and the ROC curves are shown in Figure 8. For the MR sampling scheme, there is little discernible difference between **FRK** v2, **INLA**, **mcmc**, and **spNNGP**. However, as one may expect upon viewing the predictions in Figure 6, **spBayes** performs poorly in comparison to the other packages due to the small number of knots it is able to employ.

The task of prediction over a completely unobserved region is challenging, and so it is no surprise that the diagnostics for the MB sampling scheme are worse than for the MR sampling scheme. In this case, we see **FRK** v2, **INLA**, and **spNNGP** performing slightly better than **mcmc**, which in turn performs better than **spBayes**. These results reflect the fact that purely covariance or nearest-neighbour based spatial models are only useful for prediction locations that are ‘close’ to data points. Spatial models that include covariates are typically much more effective at predicting over regions of data paucity.

Overall, these results suggest that **FRK** v2 is comparable to, or favourable to, other packages in this application where point-referenced spatial data are featured. However, the main advantages of **FRK** v2 lie in the ease with which it does more elaborate analyses with spatial or spatio-temporal non-Gaussian data of differing support, as shown in the next sections.

4.2. Block prediction: Contaminated soil

Between the years 1954 and 1963, nuclear devices were detonated at Area 13 of the Nevada Test Site in the United States, contaminating the surrounding soil with the radioactive element americium (Am). The data set we use in this example comprises Am concentrations (in 10^3 counts per minute) in a spatial domain immediately surrounding Ground Zero (GZ, the location where the devices were detonated); it was previously analysed by [Huang, Yao, Cressie, and Hsing \(2009\)](#) and [Paul and Cressie \(2011\)](#). The total number of measurements (including some that are co-located) is 212. The left and centre panels of Figure 9 show the data on the original scale and on the log scale, respectively. [Paul and Cressie \(2011\)](#) note that the Am concentrations are clearly lognormally distributed, and that soil remediation is often done by averaging the contaminant over pre-specified spatial regions of D called blocks. Hence, this application requires lognormal prediction over blocks, a task well suited to **FRK** v2. The right panel of Figure 9 shows two blocking schemes that we predict over: Both schemes contain

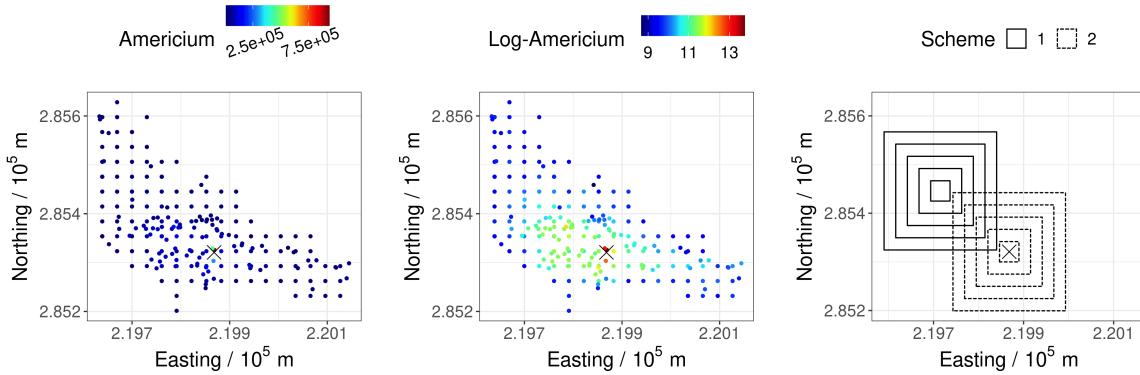


Figure 9: Americium (Am) soil data. The ‘x’ denotes Ground Zero (GZ), where the devices were detonated. (Left) Am concentrations on the original scale. (Centre) Am concentrations on the log scale. (Right) Blocking schemes: Scheme 1 (solid lines), centred away from GZ, and Scheme 2 (dashed lines), centred on GZ.

five blocks, but one scheme is centred away from GZ, and the other is centred on GZ. These blocking schemes represent the user-specified prediction regions, $\{\tilde{R}_l : l = 1, \dots, 10\}$.

As in [Paul and Cressie \(2011\)](#), we use a piece-wise linear trend in the ‘distance from GZ’ as a fixed effect. Specifically, the log of the observations within a distance of 30.48m (100 ft) from GZ are assumed to follow a different trend to those observations beyond 30.48m from GZ, making up two regimes depending on distance from GZ. In **FRK**, covariates are provided with the BAU object and, hence, in this example, we must first construct the BAUs; here, we do this using the helper function `auto_BAUs()`:

```
R> BAUs <- auto_BAUs(manifold = plane(), type = "grid", data = Am_data)
```

The following code constructs the covariates that are needed to fit this piece-wise linear trend: `BAUs$x1` and `BAUs$x3` are indicator variables used to model the intercepts in each regime, and `BAUs$x2` and `BAUs$x4` are used to model the slopes of the trend in each regime.

```
R> d_BAU <- distR(coordinates(BAUs), Ground_Zero)
R> BAUs$x1 <- d_BAU < 30.48
R> BAUs$x2 <- d_BAU * BAUs$x1
R> BAUs$x3 <- d_BAU >= 30.48
R> BAUs$x4 <- d_BAU * BAUs$x3
```

Spatial statistical modelling for this problem is done by setting `response = "Gaussian"` and `link = "log"` when calling `FRK()`. In order to mimic lognormal block kriging, which models the response as a lognormal process, here we fix the measurement-error variance to a value that is small relative to the total variance of the data (specifically, we set the measurement-error variance equal to 1, while $sd(\mathbf{Z}) = 77300$) prior to model fitting. In summary, the data model we use is Gaussian, $Z_j | \boldsymbol{\mu} \sim \text{Gau}(\mu_{Z_j}, 1)$ for $j = 1, \dots, m$, and we use a log link function, $g(\cdot) = \log(\cdot)$. Note that in the code below, we suppress the global intercept since we model the region-specific intercepts separately through `x1` and `x3`.

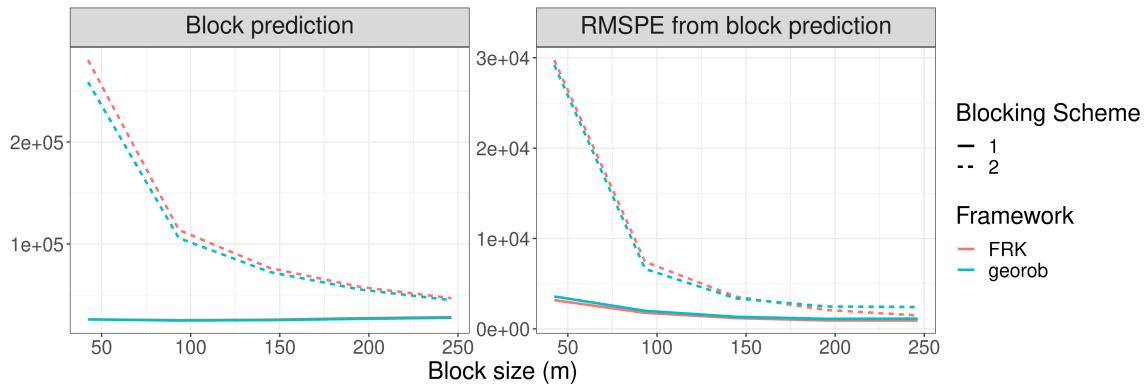


Figure 10: Plots of block-predictions (left) and RMSPE (right) of Am concentrations against block size, $|B|^{1/2}$. Both quantities are in units of 10^3 counts per minute. In both panels, the solid line corresponds to Scheme 1, and the dashed line corresponds to Scheme 2; the red line corresponds to **FRK** v2, and the blue line corresponds to **georob**.

```
R> Am_data$std <- 1
R> S <- FRK(f = Am ~ -1 + x1 + x2 + x3 + x4,
+   data = Am_data, BAUs = BAUs,
+   response = "gaussian", link = "log", est_error = FALSE)
```

In order to generate block-level predictions, we pass a ‘`SpatialPolygonsDataFrame`’ object into the `newdata` argument of `predict()`. In the following code, `blocks` is a ‘`SpatialPolygonsDataFrame`’ object containing the polygons corresponding to the two blocking schemes shown in Figure 9:

```
R> pred <- predict(S, newdata = blocks)
```

To validate the **FRK** v2 predictions, we used the R package **georob** version 0.3.14 (Papritz 2020), which implements an unbiased back-transformation of kriging predictions of log-transformed data (Cressie 2006). Computation times for kriging do not scale well with sample size, however the size of this data set is sufficiently small for straightforward kriging to be possible. The package **georob** provides users with two approaches to lognormal block kriging; we used the ‘optimal predictor’, as recommended by the **georob** manual when predicting over large blocks. Figure 10 shows the block predictions and associated RMSPE obtained using **FRK** v2 and **georob** for the two blocking schemes shown in Figure 9. It can be clearly seen that the results are very similar despite the use of dimension reduction in **FRK** v2.

4.3. Spatial change-of-support: Poverty in Sydney

The Australian Statistical Geography Standard (ASGS) defines a series of nested geographical areas in Australia known as Statistical Area Levels. Statistical Area Level 3 (SA3) regions are aggregations of Statistical Area Level 2 (SA2) regions, and SA2 regions are aggregations of Statistical Area Level 1 (SA1) regions. In this example, we consider a region of the state of New South Wales in Australia, which contains 7,909 SA1 regions, 180 SA2 regions, and 31 SA3 regions, and we aim to infer ‘poverty’ levels at the SA1 and SA3 regions from a data set

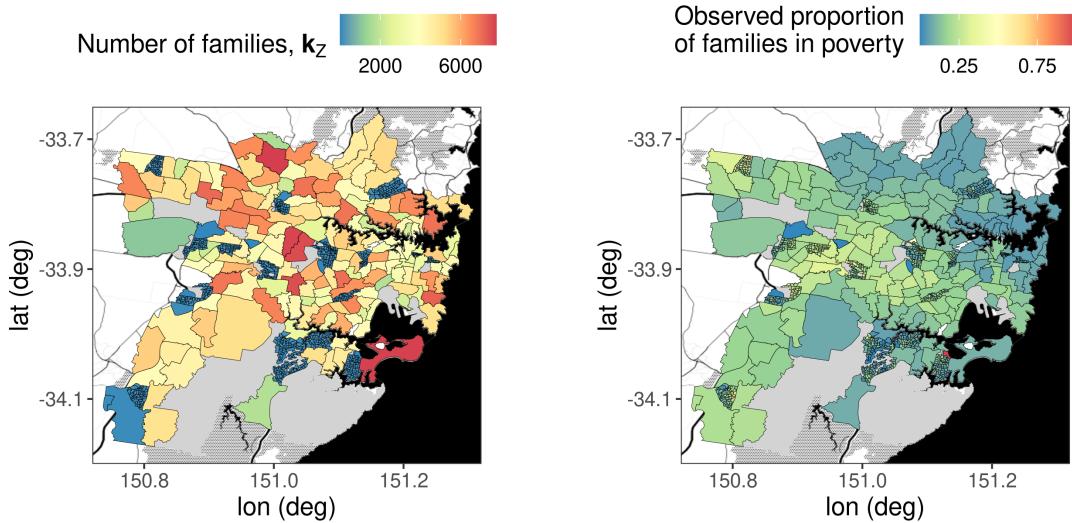


Figure 11: Training data on SA1/SA2 regions used for modelling the proportion of families ‘in poverty’ (see Appendix F for how we define ‘in poverty’). (Left) The number of families, k_Z . (Right) The observed proportion of families in poverty computed by dividing the number of families in poverty by the total number of families, that is, Z_j/k_{Z_j} , for $j = 1, \dots, m$. Solid-grey regions correspond to SA1/SA2 regions in which the total number of families is zero. The data are overlayed on a Stamen base map, where the textured grey areas correspond to bushland, large black areas correspond to ocean or large bodies of water, and solid-black lines correspond to major arterial roads. Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

containing mostly SA2 data and a small amount of SA1 data. The data were collected in the Australian Census of 2011, and they consist of the number of families of various types within a range of weekly income brackets; in Appendix F, we provide further details on the way in which we define the poverty line for each family type. Note that data at the SA1 regions are available, and we use these to validate our down-scaled predictions.

Sampling once from a large area is often relatively inexpensive compared to acquiring multiple samples from small areas. Our training data, shown in Figure 11, is reflective of such a scenario. It includes mostly SA2 regions, but some SA1 regions have also been included.

We use SA2-region (and some SA1-region) data for model fitting, and we use the SA1 regions as the BAUs. Since many of the observation supports encompass multiple BAUs (SA2 regions are aggregations of SA1 regions), the BAU-level size parameters \mathbf{k} , here the number of families in each SA1 region, are needed and must be provided with the BAU object (see Section 2.5).

```
R> SA1s$k_BAU <- SA1s$number_of_families
```

Our data model is $Z_j | \boldsymbol{\mu}, \mathbf{k}_Z \sim \text{Bin}(k_{Z_j}, \pi_{Z_j})$, for $j = 1, \dots, m$, and we use the logit link function, $f(\pi_i) = \log(\frac{\pi_i}{1-\pi_i})$ in (19). The model is established and fit in the code below.

```
R> S <- FRK(f = number_of_families_in_poverty ~ 1,
+   data = SA2s_and_some_SA1s, BAUs = SA1s,
+   response = "binomial", link = "logit")
```

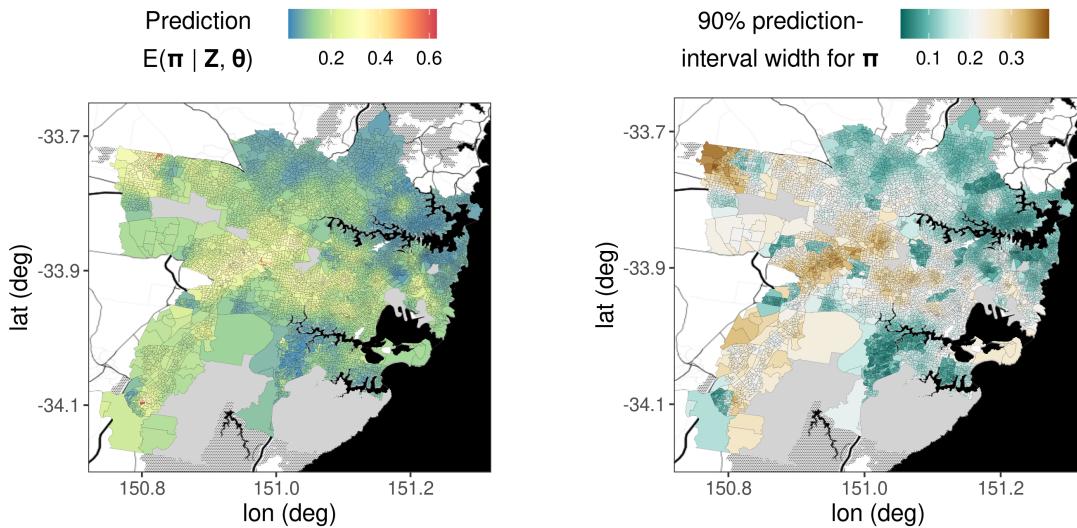


Figure 12: SA1-level predictions. (Left) Prediction of the probability process, π , representing the proportion of families in poverty in each SA1 region. (Right) The 90% prediction-interval width for each element of π . Solid-grey regions correspond to SA2 regions in which the total number of families is zero, and hence they are omitted from the study. For details on the underlying Stamen base map, refer to the caption of Figure 11.

Spatial prediction over all of the SA1 regions is obtained as follows:

```
R> SA1_predictions <- predict(S)
```

Recall that, by default, `predict()` returns predictions of the mean process, μ , and, if applicable, the probability process, π ; here, we focus on π , which can be interpreted as the proportion of families living below the poverty line. The predictions and associated uncertainty over the SA1 regions are shown in Figure 12, which was generated using `plot()`. Predicting over different spatial supports is straightforward with **FRK** v2. We predict over the SA3 regions by passing them as a ‘`SpatialPolygonsDataFrame`’ object to the argument `newdata`.

```
R> SA3_predictions <- predict(S, newdata = SA3s)
```

Figure 13 shows the SA3-region predictions and associated prediction-interval widths for the probability process, π_P . Again, this graphic was generated using `plot()`.

We assessed the model’s ability to quantify uncertainty over the SA1 regions by computing the empirical coverage from nominal 90% prediction intervals obtained via simulated predictive data at the SA1 level. We found the empirical coverage to be 90.8%, which is very close to the nominal value of 90%. Note that the inclusion of some fine-scale data (SA1 region data) greatly aids in the estimation of the fine-scale variance parameter, σ_ξ^2 , and hence in providing valid prediction intervals when downscaling. If only coarse-resolution data were available (i.e., all data supports were associated with multiple BAUs), to avoid identifiability issues, **FRK** v2 automatically fixes σ_ξ^2 before fitting the model with **TMB**, to a rough, and possibly unreliable, estimate. If one does know σ_ξ^2 or can obtain a reliable estimate of it (e.g., using past census data), one may specify it using the argument, `known_sigma2fs`.

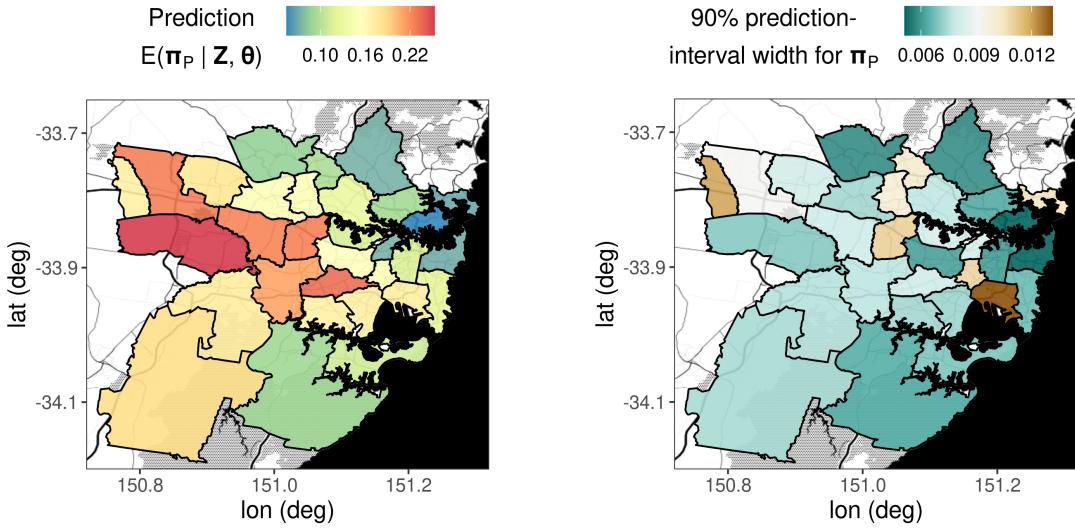


Figure 13: SA3-level predictions. (Left) Prediction of the probability process, π_P , representing the proportion of families in poverty in each SA3 region. (Right) The 90% prediction-interval width for each element of π_P . For details on the underlying Stamen base map, refer to the caption of Figure 11.

4.4. Spatio-temporal data: Crime in Chicago

The city of Chicago is divided into 77 so-called community areas (CAs). An attractive property of CAs is their relative consistency, with boundaries that have changed little since their inception in the 1920s ([The University of Chicago Library 2020](#)). In this study, we model the number of crimes in each CA between the years 2001 and 2019 inclusive. A full list of crimes committed in Chicago during this period is provided by the Chicago Police Department; the data are available at https://hpc.niasra.uow.edu.au/ckan/en_AU/dataset/chicago_crime_dataset. We considered only crimes labelled as ‘assault’ or ‘battery’; there were roughly 1.75 million such crimes in total between 2001 and 2019. We are interested in modelling the total number of crimes in each CA and in each year; once binned into CA-year bins (done automatically by **FRK**), the number of aggregated-level data is $76 \times 19 = 1444$. The CA containing O’Hare airport is non-populous and is almost disjoint from the other CAs; for simplicity, we excluded it from this analysis, leaving 76 CAs.

In this example, we use the CAs as our spatial BAUs, which are read in from a shapefile and stored as a ‘`SpatialPolygonsDataFrame`’ object. Spatio-temporal BAUs may then be constructed by passing the CAs and data into `auto_BAUs()`. In this case, the spatio-temporal BAUs are space-time volumes constructed by taking all combinations of the spatial BAU footprints with the yearly intervals that make up the 19-year period of interest.

```
R> ST_BAUs <- auto_BAUs(manifold = STplane(), data = chicago_crimes_fit,
+   spatial_BAUs = community_areas, tunit = "years")
```

When modelling crime, it is natural to include population, or population density, as a covariate. As the CAs are of unequal area, we use population rather than population density.

This covariate was obtained from the [Chicago Metropolitan Agency for Planning \(2017\)](#). It is difficult to obtain population data for every year so, for simplicity, we assume that population was constant over the time-span of the data.

Next, we generate spatio-temporal basis functions automatically using `auto_basis()`. Then, we initialise and fit the ‘SRE’ object using `FRK()`, setting `response = "poisson"` and `link = "log"` in order to use a Poisson data model, $Z_j | \boldsymbol{\mu} \sim \text{Poi}(\mu_{Z_j})$, for $j = 1, \dots, m$, and a log link function, $g(\cdot) = \log(\cdot)$. Setting `sum_variables = "number_of_crimes"` indicates that we wish to sum (rather than average) the individual crimes into aggregated-level data representing the total count in each CA and each year. As the number of spatial BAUs (the CAs) is relatively low, and we have observed each spatial BAU multiple times, we may attribute to each spatial BAU its own fine-scale variance parameter by setting `fs_by_spatial_BAU = TRUE` (see Section 2.6). We excluded the years 2010 and 2019 from the training data and used them to evaluate crime predictions and forecasts, respectively.

```
R> S <- FRK(f = number_of_crimes ~ log(population),
+   data = chicago_crimes_fit, basis = basis, BAUs = ST_BAUs,
+   response = "poisson", link = "log",
+   sum_variables = "number_of_crimes", fs_by_spatial_BAU = TRUE)
```

Finally, we generate predictive data over the spatio-temporal BAUs using `predict()`, and plot the results using `plot()`.

The observed (withheld) number of crimes, predicted number of crimes, and prediction uncertainty for the prediction (2010) and forecast (2019) years are shown in Figure 14. For both years, Figure 14 shows agreement between the predicted and observed number of crimes. Furthermore, the prediction uncertainty is roughly proportional to the predicted value, as expected when counts are modelled. For the prediction year and forecast year, we also computed the empirical coverage when using 90%, 80%, 70%, and 60% prediction intervals, and the mean absolute percentage error (MAPE; see Appendix E). We consistently observed that the $x\%$ empirical coverage in the year 2010 was slightly higher (4% on average) than the $x\%$ nominal coverage, whilst it was lower (13% on average) in the forecast year. We observed MAPE scores of 4% and 9% in the years 2010 and 2019, respectively. The slightly worse results in the year 2019 are expected, as forecasting into the future is a harder task than predicting within the time span of the data. Nonetheless, these predictions/forecasts are cause for optimism given the complexity of modelling crime in a spatio-temporal setting.

Next, we next focus on three randomly-selected CAs: Archer heights, Ashburn, and Roseland. The time series of the observed data, predictions, and 90% prediction intervals for these CAs, are shown in Figure 15. The prediction intervals are slightly wider in validation years (2010 and 2019) than in observed years, and the observed (withheld) number of crimes is contained within the prediction interval for all time points for these CAs.

5. Conclusion

In this paper, we have described **FRK** v2, which is a major upgrade to **FRK** v1. Substantial enhancements allow for the spatial and spatio-temporal modelling of, and large-scale prediction from, big, non-Gaussian data sets. Using a GLMM model and the software **TMB**, **FRK** v2 can now cater for many distributions within the exponential family, as well as many

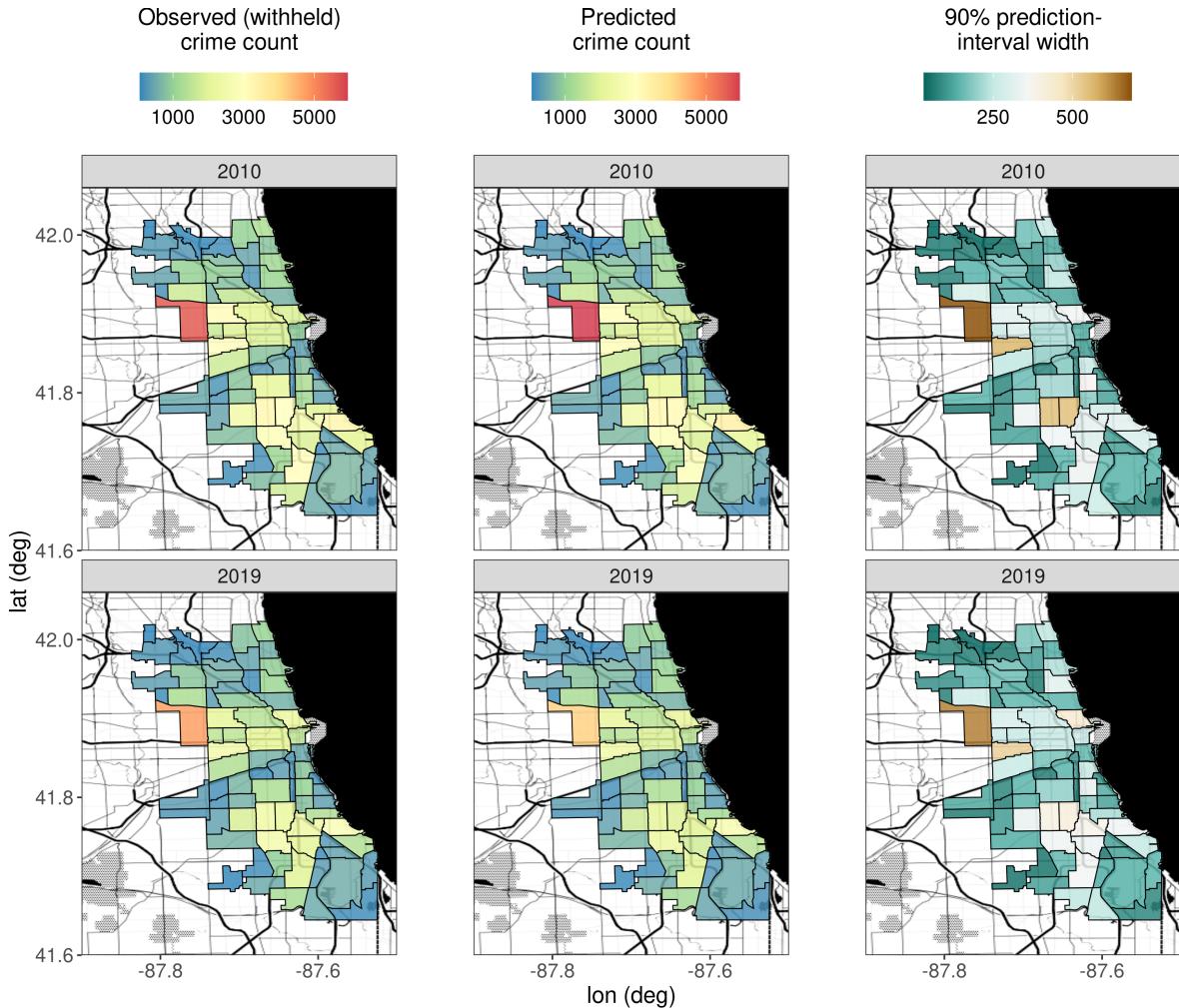


Figure 14: Observed (withheld) number of crimes, predictions, and prediction-interval width over Chicago in the prediction (2010) and forecast (2019) years. The first row corresponds to the year 2010, and the second row corresponds to the year 2019. The first column shows the observed (withheld) number of crimes; the second column shows the predicted number of crimes; and the third column shows the width of a prediction interval with a nominal coverage of 90%. For details on the underlying Stamen base map, refer to the caption of Figure 11 (the large black region in this figure correspond to Lake Michigan).

link functions. Furthermore, **FRK** v2 allows for the use of many more basis functions when modelling the spatial process, and hence it can often achieve more accurate predictions than **FRK** v1 (Table 3). The existing functionality of **FRK** v1 is retained with this extension; in particular, the package makes use of automatic basis-function construction, it is capable of handling both point-referenced and areal data, and it resolves the spatial change-of-support problem through the use of BAUs. The current version now provides a highly accessible and user-friendly approach to spatial and spatio-temporal modelling of big data in both a Gaussian and non-Gaussian setting.

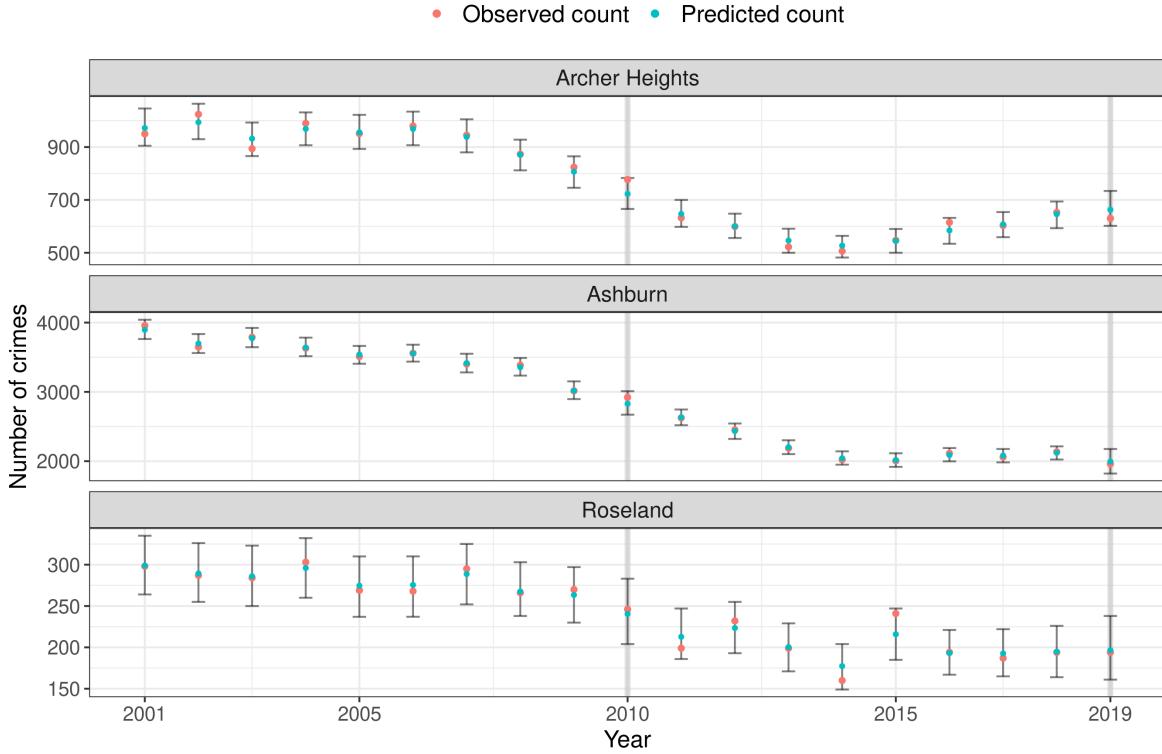


Figure 15: Time-series plots of predicted and observed number of crimes for three randomly-selected CAs. The prediction (2010) and forecast (2019) years are highlighted in light-grey. The observed number of crimes in each year is indicated by a red dot, whilst the predicted number of crimes is indicated by a blue dot. The error bars represent a 90% prediction interval. We note that the prediction intervals are slightly wider in validation years (2010 and 2019) than in observed years, and that the observed number of crimes is contained within the prediction interval in all years for these CAs.

One requirement of the framework is that covariates need to be known for every BAU, which may not be the case if covariates are recorded only at the data-support level. Spatial interpolation of the covariates can be used to address this problem. Another requirement is the practical necessity to fix the fine-scale variance parameter in spatial change-of-support applications; note that this is not an issue if one is able to obtain a reliable estimate through other means (e.g., via previously sampled data or via quality-control experiments). We are currently exploring avenues to relax this requirement via the provision of a robust offline estimate. Despite the added flexibility of **FRK** v2, several models of interest, such as the zero-inflated Poisson, are still not catered for. Future work will see the introduction of such models. Further future work includes adding support for multivariate responses, for which basis-function models are well suited (for an overview, see [Cressie et al. 2021](#), Sec. 4.1), and for spatially varying regression coefficients, which are sometimes needed in spatial analyses.

Acknowledgments

Matthew Sainsbury-Dale's research was supported by an Australian Government Research Training Program Scholarship. Andrew Zammit-Mangion's and Noel Cressie's research was supported by an Australian Research Council (ARC) Discovery Project, DP190100180. Andrew Zammit-Mangion's research was also supported by an ARC Discovery Early Career Research Award, DE180100203. The authors would like to thank Rajib Paul for providing the Americium data analysed in Section 4.2, Michael Bertolacci for his discussion surrounding the MODIS comparison study, and Yi Cao for his technical assistance during the project. We are also grateful to two anonymous reviewers whose suggestions improved our article.

References

- Bachl FE, Lindgren F, Borchers DL, Illian JB (2019). “**inlabru**: An R package for Bayesian Spatial Modelling from Ecological Survey Data.” *Methods in Ecology and Evolution*, **10**, 760–766.
- Bates D, Maechler M, Davis TA (2019). **Matrix**: Sparse and Dense Matrix Classes and Methods. R package version 1.2-17, URL <http://Matrix.R-forge.R-project.org/>.
- Bell BM (2005). “**CppAD**: A Package for C++ Algorithmic Differentiation.” <http://www.coin-or.org/CppAD>. Accessed: 2021-08-10.
- Box GEP (1980). “Sampling and Bayes' Inference in Scientific Modelling and Robustness.” *Journal of the Royal Statistical Society A*, **143**, 383 – 430.
- Bradley JR, Holan SH, Wikle CK (2018). “Computationally Efficient Multivariate Spatio-Temporal Models for High-Dimensional Count-Valued Data (With Discussion).” *Bayesian Analysis*, **13**, 253–310.
- Bradley JR, Wikle CK, Holan SH (2016). “Bayesian Spatial Change of Support for Count-Valued Survey Data with Application to the American Community Survey.” *Journal of the American Statistical Association*, **111**, 472–487.
- Bradley JR, Wikle CK, Holan SH (2019). “Spatio-Temporal Models for Big Multinomial Data Using the Conditional Multivariate Logit Beta Distribution.” *Journal of Time Series Analysis*, **50**, 363–382.
- Chicago Metropolitan Agency for Planning (2017). “Chicago Community Data Snapshots.” Technical report. URL https://www.cmap.illinois.gov/documents/10180/126764/_Combined_AllCCAs.pdf/. Accessed: 2021-08-10.
- Cox DR, Snell EJ (1968). “A General Definition of Residuals (With Discussion).” *Journal of the Royal Statistical Society B*, **30**, 248–275.
- Cressie N (1993). *Statistics for Spatial Data*. Revised edition. John Wiley & Sons, Hoboken, NJ. ISBN 0387310738.
- Cressie N (2006). “Block Kriging for Lognormal Spatial Processes.” *Mathematical Geology*, **38**, 413–443.

- Cressie N, Johannesson G (2008). “Fixed Rank Kriging for Very Large Spatial Data Sets.” *Journal of the Royal Statistical Society B*, **70**, 209–226.
- Cressie N, Sainsbury-Dale M, Zammit-Mangion A (2021). “Basis-Function Models in Spatial Statistics.” *Annual Review of Statistics and its Applications*, **9**, 373–400.
- Datta A, Banerjee S, Finley AO, Gelfand AE (2016). “Hierarchical Nearest-Neighbour Gaussian Process Models for Large Geostatistical Datasets.” *Journal of the American Statistical Association*, **111**, 800–812.
- Diggle PJ, Tawn JA, Moyeed RA (1998). “Model-Based Geostatistics.” *Journal of the Royal Statistical Society C*, **47**, 299–350.
- Dunn KP, Smyth GK (1996). “Randomized Quantile Residuals.” *Journal of Computational and Graphical Statistics*, **5**, 1–10.
- Finley AO, Banerjee S, Gelfand AE (2015). “**spBayes** for Large Univariate and Multivariate Point-Referenced Spatio-Temporal Data Models.” *Journal of Statistical Software*, **63**(13), 1–28.
- Finley AO, Datta A, Banerjee S (2020). “**spNNGP** R Package for Nearest Neighbour Gaussian Process Models.” *arXiv:2001.09111*.
- Furrer R, Nychka D, Genton MG (2006). “Covariance Tapering for Interpolation of Large Spatial Datasets.” *Journal of Computational and Graphical Statistics*, **15**, 502–523.
- Gelman A, Hill J (2007). *Data Analysis using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge, England.
- Gelman A, Meng XL, Stern H (1996). “Posterior Predictive Assessment of Model Fitness via Realized Discrepancies.” *Statistica Sinica*, **6**, 733 – 807.
- Gneiting T, Balabdaoui F, Raftery AE (2007). “Probabilistic Forecasts, Calibration and Sharpness.” *Journal of the Royal Statistical Society B*, **69**, 243–268.
- Guennebaud G, Jacob B, et al. (2010). “**Eigen** v3.” <http://eigen.tuxfamily.org>. Accessed: 2021-08-10.
- Hartig F (2022). **DHARMA: Residual Diagnostics for Hierarchical (Multi-Level/Mixed) Regression Models**. R package version 0.4.5, URL <https://CRAN.R-project.org/package=DHARMA>.
- Hastie T, Tibshirani R, Friedman J (2009). *The Elements of Statistical Learning*. Second edition. Springer-Verlag, New York.
- Heaton MJ, Datta A, Finley AO, Furrer R, Guinness J, Guhaniyogi R, Gerber F, Gramacy RB, Hammerling D, Katzfuss M, Lindgren F, Nychka DW, Sun F, Zammit-Mangion A (2019). “A Case Study Competition Among Methods for Analyzing Large Spatial Data.” *Journal of Agricultural, Biological and Environmental Statistics*, **24**, 398–425.
- Hersbach H (2000). “Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems.” *American Meteorological Society*, **15**, 559–570.

- Huang C, Yao Y, Cressie N, Hsing T (2009). “Multivariate Intrinsic Random Functions for Cokriging.” *Mathematical Geosciences*, **41**, 887–904.
- Hughes J (2014). “**ngspatial**: A Package for Fitting the Centered Autologistic and Sparse Spatial Generalized Linear Mixed Models for Areal Data.” *The R Journal*, **6**, 81–95.
- Kristensen K, Nielsen A, Berg CW, Skaug H, Bell BM (2016). “**TMB**: Automatic differentiation and Laplace approximation.” *Journal of Statistical Software*, **70**(5), 1–21.
- Lee BS, Park J (2020). “A Scalable Partitioned Approach to Model Massive Nonstationary Non-gaussian Spatial Datasets.” *arXiv:2001.09111*.
- Leroux B, Lei X, Breslow N (2000). “Estimation of Disease Rates in Small Areas: A New Mixed Model for Spatial Dependence.” In M Halloran, D Berry (eds.), *Statistical Models in Epidemiology, the Environment and Clinical Trials*, pp. 179–191. Springer-Verlag, New York.
- Lindgren F, Rue H (2015). “Bayesian Spatial Modelling with **R-INLA**.” *Journal of Statistical Software*, **63**(19), 1–25.
- Lindgren F, Rue H, Lindström J (2011). “An Explicit Link Between Gaussian fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach.” *Journal of the Royal Statistical Society B*, **73**, 423–498.
- Lopes HF, Gamerman D, Salazar E (2011). “Generalized Spatial Dynamic Factor Models.” *Computational Statistics and Data Analysis*, **55**, 1319–1330.
- McCullagh P, Nelder JA (1989). *Generalized Linear Models*. Chapman & Hall, London, UK.
- Melbourne Institute of Applied Economic and Social Research (2011). “Poverty Lines: Australia, March Quarter 2011.” Technical report. URL <https://melbourneinstitute.unimelb.edu.au/assets/documents/poverty-lines/2017/Poverty-Lines-Australia-March-Quarter-2011.pdf>. Accessed 2021-08-10.
- MODIS Characterization Support Team (2015). “MODIS 500m Calibrated Radiance Product. NASA MODIS Adaptive Processing System, Goddard Space Flight Center, USA.” <https://mcst.gsfc.nasa.gov/>.
- Nychka D, Hammerling D, Sain S, Lenssen N (2016). *LatticeKrig: Multiresolution Kriging Based on Markov Random Fields*. R package version 6.2, URL www.image.ucar.edu/LatticeKrig.
- Papritz A (2020). **georob**: Robust Geostatistical Analysis of Spatial Data. R package version 0.3-13, URL <https://cran.r-project.org/web/packages/georob/index.html>.
- Paul R, Cressie N (2011). “Lognormal Block Kriging for Contaminated Soil.” *European Journal of Soil Science*, **62**, 337–345.
- Pebesma EJ, Bivand RS (2005). “Classes and Methods for Spatial Data in R.” *R News*, **5**, 9–13.
- R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

- Rubin DR (1984). “Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician.” *Annals of Applied Statistics*, **12**, 1151 – 1172.
- Rue H, Martino S (2007). “Approximate Bayesian Inference for Hierarchical Gaussian Markov random field models.” *Journal of Statistical Planning and Inference*, **137**, 3177–3192.
- Rue H, Martino S, Chopin N (2009). “Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations.” *Journal of the Royal Statistical Society B*, **71**, 319–392.
- Sengupta A, Cressie N (2013). “Hierarchical Statistical Modelling of Big Spatial Datasets Using the Exponential Family of Distributions.” *Spatial Statistics*, **4**, 14–44.
- The University of Chicago Library (2020). “Spatially Referenced Census Data for the City of Chicago: Sources Available at or Through the University of Chicago Library.” URL <https://www.lib.uchicago.edu/e/collections/maps/censusinfo.html>. Accessed: 2021-08-10.
- Tierney L, Kadane JB (1986). “Accurate Approximations for Posterior Moments and Marginal Densities.” *Journal of the American Statistical Association*, **81**, 82–86.
- Wang C, Furrer R (2021). “Combining Heterogeneous Spatial Datasets With Process-Based Spatial Fusion Models: A Unifying Framework.” *Computational Statistics & Data Analysis*, **161**, 107240. doi:[10.1016/j.csda.2021.107240](https://doi.org/10.1016/j.csda.2021.107240).
- Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York, NY. URL <https://ggplot2.tidyverse.org>.
- Wood S (2017). *Generalized Additive Models: An Introduction with R*. Second edition. Chapman and Hall/CRC, Boca Raton, FL.
- Zammit-Mangion A, Cressie N (2021). “**FRK**: An R Package for Spatial and Spatio-Temporal Prediction with Large Datasets.” *Journal of Statistical Software*, **98**(4), 1–48.
- Zhang B, Cressie N (2020). “Bayesian Inference of Spatio-Temporal Changes of Arctic Sea Ice.” *Bayesian Analysis*, **15**, 605–631.

A. Construction of basis functions and BAUs

In this appendix, we review the construction of basis functions and BAUs with **FRK**. Further details for each function discussed below are available in the package manual.

In **FRK**, several standard basis functions, including the compactly-supported bi-square basis functions (default), may be constructed automatically from the data using `auto_basis()`, or manually using `local_basis()`. These functions produce an object of class ‘Basis’, and arbitrary, user-defined basis functions may be constructed with the constructor `Basis()`. In a spatio-temporal setting, **FRK** accommodates spatio-temporal data by using spatio-temporal basis functions constructed via a tensor product of spatial and temporal basis functions; this is achieved with the function `TensorP()`, which returns an object of class ‘TensorP_Basis’. Useful methods include `eval_basis()`, which evaluates a set of basis functions over arbitrary points or polygons, and `show_basis()`, which visualizes a set of basis functions.

The primary function for constructing BAUs is `auto_BAUs()`, which automatically constructs the BAUs as an object of class ‘`SpatialPixelsDataFrame`’ or ‘`STDFD`’, depending on whether the data is spatial or spatio-temporal. In a spatio-temporal setting, the user may provide spatial BAUs to `auto_BAUs()` via the argument `spatial_BAUs`, which can be useful when the spatial domain can be partitioned using some real-world boundaries (e.g., the example in Section 4.4). The function `BAUs_from_points()` constructs BAUs from point-level data, which can be useful for replicating traditional geostatistical analyses.

B. Parameterisations of \mathbf{K} and \mathbf{Q}

Recall from Section 2.1 that **FRK** v2 allows the covariance matrix of basis-function coefficients, $\boldsymbol{\eta}$, to be parameterised using either a covariance matrix, \mathbf{K} , or using a precision matrix, \mathbf{Q} . In this appendix, we describe the parameterisation of these matrices. Both \mathbf{K} and \mathbf{Q} are block-diagonal matrices, wherein basis-function coefficients within a basis-function resolution are dependent, but independent between different resolutions. Hence, \mathbf{K} and \mathbf{Q} are fully defined via their intra-resolution dependencies.

B.1. Covariance matrix \mathbf{K}

Let $K_k(\mathbf{s}, \mathbf{s}^*)$ denote the covariance function associated with the basis-function coefficients corresponding to the k th basis-function resolution. In **FRK**, we let $K_k(\mathbf{s}, \mathbf{s}^*)$ be the exponential covariance function, that is,

$$K_k(\mathbf{s}, \mathbf{s}^*) = \sigma_k^2 \exp \left\{ \frac{-d(\mathbf{s}, \mathbf{s}^*)}{\tau_k} \right\}, \quad (\text{B.1})$$

where $d(\mathbf{s}, \mathbf{s}^*)$ is the distance between two basis-function centroids $\mathbf{s}, \mathbf{s}^* \in D$, σ_k^2 is a variance parameter, and τ_k is a length-scale parameter. The k th sub-block of \mathbf{K} is formed by evaluating (B.1) for all pairs of basis-function centroids at the k th resolution.

Clearly (B.1) is always non-zero for $\sigma_k^2 > 0$, however it is often reasonable to assume that coefficients associated with fine-resolution basis functions separated by medium-to-large distances are uncorrelated. To increase sparsity, **FRK** v2 now allows covariance tapering (Furrer, Nychka, and Genton 2006) of the intra-resolution covariance function. Noting that (B.1) is a

special case of the Matérn covariance function with smoothness parameter $\nu = 0.5$, we follow the recommendation of [Furrer et al. \(2006\)](#) and use the spherical taper:

$$T_{\beta_k}(\mathbf{s}, \mathbf{s}^*) = \left\{ 1 - \frac{d(\mathbf{s}, \mathbf{s}^*)}{\beta_k} \right\}_+^2 \left\{ 1 + \frac{d(\mathbf{s}, \mathbf{s}^*)}{2\beta_k} \right\}, \quad (\text{B.2})$$

where $x_+ \equiv \max(0, x)$, and β_k is a resolution-dependent tapering parameter controlling the strength of the taper. In **FRK** v2, we let β_k be proportional to the minimum distance between basis-function centroids; specifically, we set $\beta_k = \text{taper} \times \text{mindist}(k)$, where $\text{mindist}(k)$ is the minimum distance between the centroids of basis functions at the k th resolution and **taper** is a user-specified argument. The tapered covariance function is obtained by taking the product of the original covariance function (B.1) and the taper function (B.2).

B.2. Precision matrix \mathbf{Q}

FRK v2 offers two types of sparse precision matrices: One is for regularly spaced basis functions, and the other is for irregularly spaced basis functions. This choice is determined by the slot **regular** in the ‘Basis’ object.

When the basis functions are regularly spaced (**regular** = TRUE), **FRK** v2 uses a precision matrix based on the Leroux model ([Leroux, Lei, and Breslow 2000](#)). Let $\mathcal{N}_{i,k}$ denote the set of first-order horizontal and vertical neighbouring basis functions of the i th basis function of resolution k , and let \mathbf{Q}_k denote the precision matrix of the basis-function coefficients at resolution k . We model the elements of \mathbf{Q}_k as

$$\{\mathbf{Q}_k\}_{i,j} = \begin{cases} \kappa_k + \rho_k |\mathcal{N}_{i,k}| & i = j \\ -\rho_k & j \in \mathcal{N}_{i,k} \\ 0 & \text{otherwise} \end{cases}, \quad (\text{B.3})$$

where κ_k and ρ_k are parameters that are estimated. We note that \mathbf{Q}_k is diagonally dominant, and hence it is positive-definite. This formulation implies that the coefficient of a given basis function is conditionally independent of all other basis-function coefficients given the coefficients of its first-order vertical and horizontal neighbours. Note that **LatticeKrig** ([Nychka, Hammerling, Sain, and Lenssen 2016](#)) uses $\mathbf{Q}_k^\top \mathbf{Q}_k$ as the precision matrix blocks, whereas **FRK** v2 uses \mathbf{Q}_k .

To cater for irregularly-spaced basis functions, **FRK** v2 also offers a sparse precision matrix based on the distance between basis-function centroids:

$$\{\mathbf{Q}_k\}_{i,j} = \begin{cases} \kappa_k - \sum_{l \neq i} \{\mathbf{Q}_k\}_{i,l} & i = j \\ -\rho_k \exp \left\{ \frac{-d(\mathbf{s}_{i,k}, \mathbf{s}_{j,k})}{\tau_k} \right\} T_{\beta_k}(\mathbf{s}_{i,k}, \mathbf{s}_{j,k}) & i \neq j \end{cases}, \quad (\text{B.4})$$

where κ_k , ρ_k , and τ_k are parameters that are estimated, and $T_{\beta_k}(\cdot, \cdot)$ is defined as in Appendix B.1. Again, this matrix is diagonally dominant and hence is positive-definite. This formulation implies that the partial correlation between basis-function coefficients decays exponentially with distance until a point (controlled by the tapering parameter β_k) at which the basis-function coefficients are conditionally independent.

C. Incidence matrices: \mathbf{C}_Z and \mathbf{C}_P

Recall from Section 2.2 that \mathbf{C}_Z aggregates the BAU-level mean process, $\boldsymbol{\mu}$, over the observation supports and, depending on the weights in (4), it can correspond to a weighted average or a weighted sum over the BAUs. In **FRK** v2, the weights w_{ij} may be controlled through the argument `normalise_wts` and the `wts` field of the ‘`SpatialPixelsDataFrame`’/‘`SpatialPolygonsDataFrame`’ object (Pebesma and Bivand 2005) used to store the BAUs. Specifically, the `wts` field allows one to attribute each BAU to a *relative* weight v_i , $i = 1, \dots, N$, such that $w_{ij} \propto v_i$, where the constant of proportionality can vary with j . For example, if the BAUs are of unequal area, then one may wish to set $v_i = |A_i|$. By default (and implicit in **FRK** v1), each v_i is set to 1. The argument `normalise_wts` controls whether \mathbf{C}_Z corresponds to a weighted sum or a weighted average. If set to `FALSE`, then $w_{ij} = v_i$ for all j (weighted sum); if set to `TRUE` (default and implicit in **FRK** v1), then the $\{w_{ij}\}$ are normalised so that each row of \mathbf{C}_Z sums to 1 (weighted average) and $w_{ij} = v_i / \sum_{l \in c_j} v_l$. Note that if $v_i = |A_i|$ and `normalise_wts = TRUE`, the normalised weights are $w_{ij} = |A_i| / |B_j|$, since the BAUs are disjoint and $\sum_{i \in c_j} |A_i| = |B_j|$.

Recall from Section 2.4 that \mathbf{C}_P aggregates the BAU-level mean process, $\boldsymbol{\mu}$, over the prediction regions and, depending on the weights in (17), it can correspond to a weighted average or a weighted sum over the BAUs. Like \mathbf{C}_Z , the relative weights, $\{\tilde{v}_i : i = 1, \dots, N\}$, such that $\tilde{w}_{ik} \propto \tilde{v}_i$, are controlled by the `wts` field of the BAU object, and the argument `normalise_wts` is used to control whether \mathbf{C}_P represents a weighted sum or a weighted average. For consistency between the model fitting and prediction stages, **FRK** v2 enforces the use of the same relative weights, $\tilde{v}_i = v_i$ for $i = 1, \dots, N$, and the same setting of `normalise_wts`, in construction of both \mathbf{C}_Z and \mathbf{C}_P .

Recall from Section 2.5 that in most applications that consider binomial or negative-binomial data models, the conditional mean of an observation is treated as a simple aggregate of the underlying mean process. Therefore, with these distributions, **FRK** v2 enforces the matrix \mathbf{C}_Z in (4) to be constructed with the relative weights $\{v_i = 1 : i = 1, \dots, N\}$ and with `normalise_wts = FALSE`, and hence $w_{ij} = 1$ in (4). Then, the mapping from the BAU-level mean, $\boldsymbol{\mu}$, to the data-level mean, $\boldsymbol{\mu}_Z$, in (5) is a simple, unweighted summation over the BAUs. Since **FRK** v2 enforces the use of the same relative weights, $\tilde{v}_i = v_i$ for $i = 1, \dots, N$, and the same setting of `normalise_wts`, in construction of both \mathbf{C}_Z and \mathbf{C}_P , the mapping from $\boldsymbol{\mu}$ to $\boldsymbol{\mu}_P$ in (18) is also a simple, unweighted summation over the BAUs.

D. Distributions with size parameters: Linking π to $\boldsymbol{\mu}$

Recall from Section 2.5 that $h(\cdot; \cdot)$ is a function that links the probability process, $\pi(\cdot)$, to the mean process, $\mu(\cdot)$. In this appendix, we give its derivation. The expectation of a binomial random variable, $Z | \pi, k \sim \text{Bin}(k, \pi)$, is $\mathbb{E}(Z) = k\pi$, which motivates the use of

$$h(\mu; k) = \frac{\mu}{k}, \quad (\text{D.1})$$

when the response distribution is binomial. The expectation of a negative-binomial random variable, $Z | \pi, k \sim \text{NB}(k, \pi)$, is $\mathbb{E}(Z) = \frac{k(1-\pi)}{\pi}$, which motivates the use of

$$h(\mu; k) = \frac{k}{\mu + k}, \quad (\text{D.2})$$

when the response distribution is negative-binomial. By construction, using $h(\cdot; \cdot)$ to link the probability process to the mean process in (20), ensures that the range of the mean process is appropriate for modelling the expectation of a binomial distributed random variable, where the mean must lie in the range $(0, k)$, or a negative-binomial distributed random variable, where the mean must lie in the range $(0, \infty)$.

E. Scoring rules

Suppose that we have a discrete validation domain $D^* \subset D$, which is used for model validation. As prediction-performance measures for the examples in this paper, we considered the following diagnostics (for simplicity, we describe the diagnostics in terms of prediction of the continuous mean process):

- (Empirical) root-mean-squared prediction error (RMSPE): Let $\hat{\mu}(\mathbf{s})$ denote a point-predictor of $\mu(\mathbf{s})$, where $\mu(\mathbf{s})$ is the true value of the mean process evaluated at $\mathbf{s} \in D^*$. Then the empirical RMSPE, used to assess point-wise predictive performance, is

$$\text{RMSPE} \equiv \sqrt{\frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} (\hat{\mu}(\mathbf{s}) - \mu(\mathbf{s}))^2}.$$

- (Empirical) mean-absolute error (MAE): Also used to assess point-wise predictive performance, the empirical MAE is

$$\text{MAE} \equiv \frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} |\hat{\mu}(\mathbf{s}) - \mu(\mathbf{s})|.$$

- (Empirical) mean-absolute percentage error (MAPE): This is similar to the empirical MAE, but considers relative error instead; the empirical MAPE is

$$\text{MAPE} \equiv \frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} \left| \frac{\hat{\mu}(\mathbf{s}) - \mu(\mathbf{s})}{\mu(\mathbf{s})} \right|.$$

- (Averaged) continuous ranked probability score (CRPS; [Gneiting, Balabdaoui, and Raftery 2007](#), sec 4.2.): The averaged CRPS is used to evaluate the predictive cumulative distribution function (CDF) of the mean process, $F(\mu; \mathbf{s}, \mathbf{Z})$, over all $\mathbf{s} \in D^*$, and is defined as

$$\text{CRPS} \equiv \frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} \int_{-\infty}^{\infty} (F(x; \mathbf{s}, \mathbf{Z}) - \mathbb{1}\{x \geq \mu(\mathbf{s})\})^2 dx,$$

where $\mathbb{1}\{\cdot\}$ denotes an indicator function that takes the value 1 if its argument is true, and 0 otherwise. For some predictive CDFs (in particular, the Gaussian and log-Gaussian), there exist closed-form expressions to compute the CRPS. However, in general, no closed-form expression exists, in which case we may use an *empirical* predictive CDF from a sample (e.g., a Monte Carlo sample) to evaluate the CRPS in terms of the respective order statistics ([Hersbach 2000](#)).

- (Averaged) interval score (IS; Gneiting *et al.* 2007, sec. 6.2): Given a set of purported $(1 - \alpha) \times 100\%$ prediction intervals for $\mu(\mathbf{s})$, $\mathbf{s} \in D^*$, the averaged IS is defined as

$$\text{IS}_\alpha \equiv \frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} \left(U(\mathbf{s}) - L(\mathbf{s}) + \frac{2}{\alpha} (L(\mathbf{s}) - \mu(\mathbf{s})) \mathbb{1}\{\mu(\mathbf{s}) < L(\mathbf{s})\} + \frac{2}{\alpha} (\mu(\mathbf{s}) - U(\mathbf{s})) \mathbb{1}\{\mu(\mathbf{s}) > U(\mathbf{s})\} \right),$$

where $L(\mathbf{s})$ and $U(\mathbf{s})$ are the lower and upper bounds of the prediction interval at location \mathbf{s} . The IS rewards narrow prediction intervals and penalises instances in which an observation misses the interval, with the size of the penalty depending on α .

- (Empirical) coverage: The empirical coverage of the prediction intervals is defined as

$$\text{Cvg} \equiv \frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} \mathbb{1}\{L(\mathbf{s}) \leq \mu(\mathbf{s}) \leq U(\mathbf{s})\}$$

If the intervals are indeed $(1 - \alpha) \times 100\%$ prediction intervals throughout D^* , the empirical coverage should be approximately equal to $1 - \alpha$.

- Brier score (Gneiting *et al.* 2007, Sec. 3): The Brier score is defined as

$$\text{Brier Score} \equiv \frac{1}{|D^*|} \sum_{\mathbf{s} \in D^*} (Z_\mathbf{s} - \hat{\pi}(\mathbf{s}))^2,$$

where $Z_\mathbf{s}$ denotes the validation datum at \mathbf{s} (taking a value of 0 or 1), and $\hat{\pi}(\mathbf{s})$ denotes a point-prediction of the probability process at \mathbf{s} .

F. Sydney poverty lines

Here we provide some details on how we define the poverty lines for the data in Section 4.3. We base our definitions of poverty lines on a Melbourne Institute of Applied Economic and Social Research (MIAESR) report that was published in March 2011 ([Melbourne Institute of Applied Economic and Social Research 2011](#)). However, the family units in the 2011 Australian Census do not align exactly with those used by the MIAESR and, since this example is shown for purely illustrative purposes, we make several assumptions. First, we assume ‘families with children’ in the Census data consist of exactly two parents and two children. Second, since ‘other families’ in the Census is difficult to interpret and categorise appropriately in the context of the MIAESR guidelines, we exclude ‘other families’ from the study (less than 2% of all families). Third, the Census data do not provide exact income figures, but rather they provide income brackets of width \$200; we thus round the MIAESR guidelines to the nearest \$200. Fourth, the Census data do not make clear whether the head of the family is in the workforce; we therefore assume that the head of the family *is* in the workforce, and hence we use the first half of Table 1 of the MIAESR report guidelines for defining poverty lines. These assumptions lead us to define poverty lines (in Australian dollars) for each family unit considered in this study as weekly incomes of: \$600 for a couple with no children, \$800 for a

couple with children, and \$600 for a one-parent family. The proportion of families we deem to be in poverty is based on their being below these thresholds.

Affiliation:

Matthew Sainsbury-Dale

National Institute for Applied Statistics Research Australia (NIASRA)

School of Mathematics and Applied Statistics

University of Wollongong

Wollongong, Australia

E-mail: msainsburysdale@gmail.com

URL: <https://github.com/msainsburysdale>