Modelling, Fitting, and Prediction with Non-Gaussian Spatial and Spatio-Temporal Data using FRK

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

Abstract

Non-Gaussian spatial and spatial-temporal data are becoming increasingly prevalent, arising from studies as far apart as small-area demographics (counts) and global remote sensing (radiant energies). **FRK** is an R package for spatial/spatio-temporal linear modelling and prediction with very large data sets. In this paper, we take **FRK** to the next level where non-Gaussian data are analysed in a generalised linear mixed model framework. The existing functionality of **FRK** is retained with this advance; in particular, it makes use of automatic basis-function construction, it can handle both point-referenced and areal data simultaneously, and it predicts process values at any spatial support from these data. These non-linear, non-Gaussian models are fitted by using the Laplace approximation and the software **TMB** to obtain likelihood-based estimates. We demonstrate innovative features in **FRK** and compare it to alternative packages, using both simulated and real data sets.

Keywords: non-Gaussian, spatial, spatio-temporal, big data, change of support, areal data, basis functions, R.

1. Introduction

Non-Gaussian spatial and spatio-temporal data arise from a vast array of sources, including statistical studies in contaminated soil (Paul and Cressie 2011), global remote sensing (Sengupta and Cressie 2016), small-area demographics (Bradley, Wikle, and Holan 2016), and earthquake magnitudes (Hu and Bradley 2018). The statistical modelling of this data is pertinent, as accurate predictions, and uncertainty quantification of those predictions, assist individuals in giving informed answers to real-world problems. There are, by now, several approaches to statistical modelling and spatial prediction with non-Gaussian data, which we review in the following paragraphs.

One widespread method to deal with non-Gaussian data is *trans-Gaussian kriging* (Cressie 1993, pg. 137–138), in which standard kriging (i.e., spatial optimal linear prediction) is used after applying a non-linear transformation to the data, and predictions are made back 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; Diggle, Tawn, and Moyeed 1998), whereby the response distribution is assumed to be a member of the exponential family, and the conditional mean is modelled using a transformation of some latent spatial process

$Y(\cdot)$. Classical models for $Y(\cdot)$ typically entail the inversion of an $n \times n$ covariance matrix, where n is the number of observations. Since this task is in general $O(n^3)$ in computational complexity, some form of dimension-reduction is required in 'big data' settings.

Reduced-rank variants of trans-Gaussian kriging are relatively under-developed (see Cressie et al. (2021, sec. 4.1) for discussion), however many modellers have used reduced-rank variants of the spatial GLMM. For instance, within the spatial GLMM framework, 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. A popular reduced-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 *et al.* (2016) use it in the spatial GLMM context. Finley, Datta, and Banerjee (2020) modelled binomial data using a spatial GLMM with a nearest neighbour Gaussian process (NNGP; Datta, Banerjee, Finley, and Gelfand 2016) assumed for $Y(\cdot)$. Lee and Park (2020) took the spatial partitioning route, where the spatial domain was partitioned 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 local processes, assumed to be independent of each other. The reduced-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 model fitting with non-Gaussian data is quite limited; we review these in the paragraph that follows.

Software packages that straightforwardly facilitate the modelling of non-Gaussian spatial and spatio-temporal data include **ngspatial** (Hughes 2014), **spBayes** (Finley, Banerjee, and Gelfand 2015), **mgcv** (Wood 2017), **spNNGP** (Finley *et al.* 2020), and **georob** (Papritz 2020). Individually, these packages all have one or more major limitations: **spBayes**, **mgcv**, and **spNNGP** are limited to point-referenced data; **spBayes** uses basis functions that depend on covariance-function parameters, so computationally it can only handle a small number of predictive-process knots, and thus causes a high degree of smoothing; **georob** is not designed for large data sets; **ngspatial**, **spBayes**, **spNNGP**, and **georob** are restricted to the spatial setting, and cater for only a small number of non-Gaussian distributions. Further, these software packages do not cater for spatial change-of-support. Some general purpose packages, like **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. A project aimed at facilitating spatial statistical modelling using **INLA** is the **inlabru** package (Bachl, Lindgren, Borchers, and Illian 2019), although spatio-temporal modelling was not implemented at the time of writing.

FRK, introduced by Zammit-Mangion and Cressie (2021), is an R package for spatial/spatio-temporal statistical modelling and prediction. The main purpose of this article is to present a major upgrade to version 2, which caters for many distributions within the exponential family using the spatial GLMM framework; we henceforth refer to it as **FRK** v2. It provides a unifying framework that handles very large, spatial and spatio-temporal non-Gaussian data, and it seamlessly ingests point-referenced and area-referenced data to solve spatial change-of-support

problems. User-friendliness is a central focus of the package: Challenging statistical problems may be tackled with only several lines of intuitive, readable code. Optimal spatial prediction proceeds through the use of an *empirical* hierarchical 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 on the surface of a sphere, a feature not offered by 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 when modelling the spatial process. Therefore, in a Gaussian setting, it often achieves more accurate predictions than previous versions of the package.

The remainder of the paper is organised as follows. In Section 2, we establish the statistical framework for **FRK** v2, and describe model fitting and prediction using the R package **TMB** (Kristensen, Nielsen, Berg, Skaug, and Bell 2016). In Section 3, we discuss the new functionalities in **FRK** v2, provide illustrative examples using simulated data, and demonstrate how using an increased number of basis functions can substantially improve predictive performance. 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 statistical model used in **FRK** v2 is a spatial GLMM, a 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 present our approach 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 previous versions of the package (henceforth referred to as **FRK** v1), as described by Zammit-Mangion and Cressie (2021). Note that here we consider the spatial case only; the extension to a spatio-temporal setting is given in Section 2.5.

In **FRK** v1/v2, we denote the latent spatial process as $Y(\cdot) \equiv \{Y(\boldsymbol{s}) \colon \boldsymbol{s} \in D\}$, where $\boldsymbol{s}$ indexes space in the spatial domain of interest D. The model for the latent process is

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

where each term in (1) is intended to capture a different form of spatial variability. First, spatially referenced covariates $\boldsymbol{t}(\cdot)$, and the associated regression parameters $\boldsymbol{\alpha}$, capture spatial variation that is linked to known, usually large-scale, explanatory variables that are elements of $\boldsymbol{t}(\cdot)$; 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. Accounting for only large and medium-to-small-scale spatial variation can result in overly smooth prediction surfaces, which in turn may lead to overly optimistic predictions;

this problem is alleviated by including a fine-scale-variation random process, $\xi(\cdot)$, which is 'almost' uncorrelated.

The medium-to-small-scale term $v(\cdot)$ is constructed as a linear combination of r spatial basis functions with random coefficients. Specifically,

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

where $\boldsymbol{\eta} \equiv (\eta_1, \ldots, \eta_r)^\top$ is an r-dimensional vector of random coefficients for the r-dimensional vector $\boldsymbol{\phi}(\boldsymbol{s}) \equiv (\phi_1(\boldsymbol{s}), \ldots, \phi_r(\boldsymbol{s}))^\top$ of pre-specified spatial basis functions evaluated at location $\boldsymbol{s} \in D$. See Zammit-Mangion and Cressie (2021) for details on how these basis functions are constructed. The fine-scale term, $\{\xi(\boldsymbol{s}) : \boldsymbol{s} \in D\}$, is modelled as a white-noise process.

FRK v1/v2 discretises the domain of interest D into N small, non-overlapping basic areal units (BAUs) $\{A_i : i = 1, \ldots, N\}$ such that $D = \cup_{i=1}^{N} A_i$. BAUs are a key element of **FRK** v1/v2, as they provide a framework that allows one to use both point-referenced and areal data simultaneously, and one that facilitates solutions to spatial change-of-support problems. Define the latent spatial process $Y(\cdot)$ evaluated over the BAUs as

$$Y_i \equiv Y(A_i);\ i = 1, \ldots, N.$$

Then, after discretisation via the BAUs, we obtain the vectorised version of (1):

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

where $\boldsymbol{Y} \equiv (Y_i : i = 1, \ldots, N)^\top$ is an N-dimensional vector corresponding to the BAUs, $\boldsymbol{T}$ and $\boldsymbol{S}$ are known design matrices, and $\boldsymbol{\xi}$ is the vector associated with the fine-scale process.

As in **FRK** v1, the elements of $\boldsymbol{\xi}$ are modelled as independent and identically distributed Gaussian random variables with variance σ^2_ξ, and $\boldsymbol{\eta}$ is modelled as a mean-zero multivariate-Gaussian random variable with covariance matrix $\boldsymbol{K}$. In **FRK** v2, $\mathrm{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$ is modelled either as $\boldsymbol{K}$ or as $\boldsymbol{Q}^{-1}$, where $\boldsymbol{Q}$ is a precision matrix. Both formulations use block-diagonal matrices, so that basis-function coefficients between resolutions are independent, and see Appendix A for details on the intra-resolution dependencies. In a non-Gaussian setting (when **TMB** is used for model fitting) we typically recommend using $\boldsymbol{Q}$ for computational reasons.

Following standard generalised linear model theory (McCullagh and Nelder 1989), **FRK** v2 uses a link function, $g(\cdot)$, to model $Y(\cdot)$ as a transformation of the mean process, $\mu(\cdot)$:

$$g\left(\mu(\boldsymbol{s})\right) = Y(\boldsymbol{s}); \quad \boldsymbol{s} \in D.$$

Then the mean process evaluated over the BAUs is

$$\mu_i = g^{-1}(Y_i);\ i = 1, \ldots, N,$$

where $g^{-1}(\cdot)$ is the inverse link function, and we define $\boldsymbol{\mu} \equiv (\mu_i : i = 1, \ldots, N)^\top$.

2.2. The data layer

Given m observations with (possibly overlapping) spatial support defined on one or more BAUs, we write the observation supports as $B_j \equiv \cup_{i \in c_j} A_i$, for $j = 1, \ldots, m$, where c_j is a

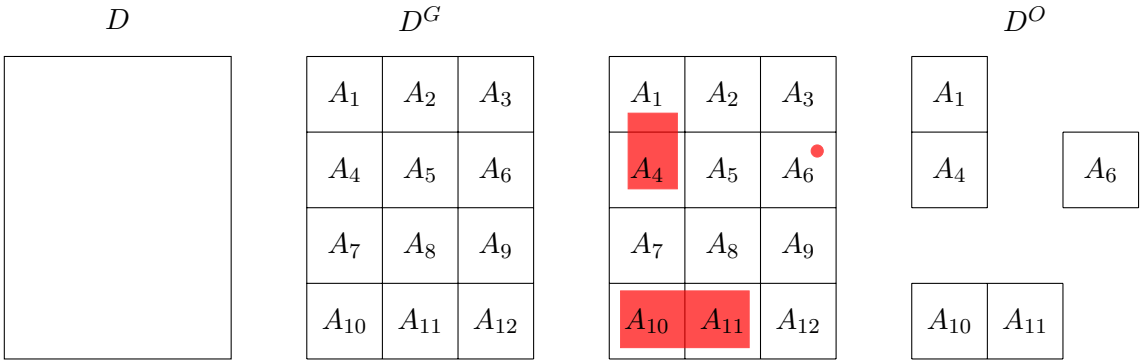

Figure 1: A simple example demonstrating how the continuous spatial domain, D, is discretised into BAUs, and how the observation domain, D^O, is derived from the observations. (Left panel) The continuous spatial domain, D. (Centre-left panel) The spatial domain discretised into $N = 12$ BAUs. (Centre-right panel) The discretised domain after observing $m = 3$ observations, two of which are area-referenced, and one that is point-referenced. (Right panel) The observation domain, D^O. The observation supports that comprise D^O are; $B_1 \equiv A_1 \cup A_4$ with $c_1 \equiv \{1, 4\}$; $B_2 \equiv A_6$ with $c_2 \equiv \{6\}$; and $B_3 \equiv A_{10} \cup A_{11}$ with $c_3 \equiv \{10, 11\}$.

non-empty set in the power set of $\{1, \ldots, N\}$ that gives the indices of the BAUs associated with observation j. We define $D^O \equiv \cup_{i=1}^{m} B_j$. The vector of observations (the data vector) is then $\boldsymbol{Z} \equiv (Z_1, \ldots, Z_m)^\top$, where $Z_j \equiv Z(B_j)$, for $j = 1, \ldots, m$. In practice, B_j may not include entire BAUs; in this case, we assume that an areal spatial support contains a BAU if and only if there is some overlap between the BAU and the spatial support. **FRK** v1 assumed that a spatial support contains a BAU if and only if the BAU centroid lies within the region; the relaxed condition in **FRK** v2 is intended to cater for non-convex BAUs, such as those used in Section 4.3, where the centroid of a given BAU may lie outside of the BAU boundary. Attributing point-referenced data to BAUs is straightforward. Figure 1 shows a simple example demonstrating how D is discretised into BAUs, and how D^O is derived from the observations.

Define the observation level mean as $\boldsymbol{\mu}_Z \equiv (\mathbb{E}(Z_1), \ldots, \mathbb{E}(Z_m))^\top$. Then since each $B_j \in D^O$ is either a BAU or a union of BAUs, one can construct an $m \times N$ matrix

$$\boldsymbol{C}_Z \equiv \Big(w_{i,j} \mathbb{I}(i \in c_j) : i = 1, \ldots, N; j = 1, \ldots, m \Big), \tag{3}$$

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

$$\boldsymbol{\mu}_Z = \boldsymbol{C}_Z \boldsymbol{\mu}. \tag{4}$$

Note that Zammit-Mangion and Cressie (2021) applied $\boldsymbol{C}_Z$ directly to $\boldsymbol{Y}$; with an identity link function, implicit in **FRK** v1, $\boldsymbol{\mu}$ and $\boldsymbol{Y}$ are equivalent. In **FRK** v2, the weights $w_{i,j}$ may be controlled through the `wts` field of the BAUs object and the argument `normalise_wts`. Specifically, the `wts` field allows one to attribute each BAU with some proportionality constant, $\{v_i : i = 1, \ldots, N\}$, such that $w_{i,j} \propto v_i$. 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, so that $w_{i,j}$ is constant for a given observation support j. The argument `normalise_wts` controls whether $\boldsymbol{C}_Z$ corresponds to a weighted sum or a weighted average; if it is `TRUE`

(default, and implicit in **FRK** v1), then the weights $w_{i,j}$ are normalised so that each row of $\boldsymbol{C}_Z$ sums to 1, and the mapping represents a weighted average. Note that if $v_i = |A_i|$ and `normalise_wts = TRUE`, the jth row sum of $\boldsymbol{C}_Z$ is $\sum_{i \in c_j} |A_i| = |B_j|$, so that the normalised weights are $w_{i,j} = |A_i|/|B_j|$.

Denoting the mean of Z_j, that is, the jth element of $\boldsymbol{\mu}_Z$, by $\{\boldsymbol{\mu}_Z\}_j$, we assume that

$$[Z_j \mid \mu(\cdot), \psi] = \mathrm{EF}(\{\boldsymbol{\mu}_Z\}_j, \psi), \tag{5}$$

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, and note that $\psi = 1$ for some distributions in the exponential family (e.g., binomial, negative-binomial, and Poisson distributions). Equation (5) implies that a given observation depends only on the value of the mean process at the corresponding observation support, rather than on the process over the whole domain. As a result, conditional on the latent spatial process, all observations are conditionally independent:

$$[\boldsymbol{Z} \mid \mu(\cdot), \psi] = [\boldsymbol{Z} \mid \boldsymbol{\mu}_{\boldsymbol{Z}}, \psi] = \prod_{j=1}^{m} \mathrm{EF}(\{\boldsymbol{\mu}_Z\}_j, \psi).$$

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

$$\ln[\boldsymbol{Z} \mid \boldsymbol{\mu}_Z, \psi] = \sum_{j=1}^{m} \left\{ \frac{Z_j \lambda(\{\boldsymbol{\mu}_Z\}_j) - b(\lambda(\{\boldsymbol{\mu}_Z\}_j))}{a(\psi)} + c(Z_j, \psi) \right\}, \tag{6}$$

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

Note that two distributions catered for by **FRK** v2, namely the binomial and negative-binomial distributions, have a known constant 'size' parameter, k_j, and a 'probability of success' parameter, π_j, associated with each datum, Z_j; see Appendix B for details on how we link the latent process $Y(\cdot)$ to the mean process $\mu(\cdot)$ when these distributions are chosen.

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

$$Z_j \mid \{\boldsymbol{\mu}_Z\}_j, \psi \overset{\text{ind}}{\sim} \mathrm{EF}(\{\boldsymbol{\mu}_Z\}_j, \psi), \quad j = 1, \ldots, m, \tag{7}$$

$$\boldsymbol{\mu}_Z = \boldsymbol{C}_Z \boldsymbol{\mu}, \tag{8}$$

$$g(\boldsymbol{\mu}) = \boldsymbol{Y}, \tag{9}$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{\alpha} + \boldsymbol{S}\boldsymbol{\eta} + \boldsymbol{\xi}, \tag{10}$$

$$\boldsymbol{\eta} \mid \boldsymbol{\vartheta} \sim \mathrm{Gau}(\boldsymbol{0}, \boldsymbol{Q}^{-1}), \tag{11}$$

$$\boldsymbol{\xi} \mid \sigma_\xi^2 \sim \mathrm{Gau}(\boldsymbol{0}, \sigma_\xi^2 \boldsymbol{V}). \tag{12}$$

where $\boldsymbol{V}$ is a known, positive-definite diagonal matrix 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 2.5. Note that **FRK** v2 is backward compatible: An identity link function and a Gaussian data model yields the model used in **FRK** v1.

2.3. Estimation

We now derive the likelihood functions required for model fitting, describe the intractable integrals that arise when using non-Gaussian data models are dealt with, and describe how **TMB** (Kristensen *et al.* 2016) is used to obtain estimates of the parameters/fixed effects, and predictions of the random effects.

Noting that $\boldsymbol{\mu}_Z$ is, through (8)–(10), completely determined by $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, and $\boldsymbol{\xi}$, the complete-data likelihood function for our model is

$$L(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}) \equiv [\boldsymbol{Z}, \boldsymbol{\eta}, \boldsymbol{\xi} \mid \boldsymbol{\theta}] = [\boldsymbol{Z} \mid \boldsymbol{\mu}_Z, \psi][\boldsymbol{\eta} \mid \boldsymbol{\vartheta}][\boldsymbol{\xi} \mid \sigma^2_\xi], \tag{13}$$

where $\boldsymbol{\theta} \equiv (\boldsymbol{\alpha}^\top, \boldsymbol{\vartheta}^\top, \sigma^2_\xi, \psi)^\top$ and $\boldsymbol{\vartheta}$ denotes the variance components associated with either $\boldsymbol{K}$ or $\boldsymbol{Q}$. The complete-data log-likelihood function, $l(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{\eta}, \boldsymbol{\xi})$, is simply the logarithm of (13). Under our modelling assumptions (7)–(12), the conditional density functions $[\boldsymbol{\eta} \mid \boldsymbol{\vartheta}]$ and $[\boldsymbol{\xi} \mid \sigma^2_\xi]$ are invariant to the specified link function and the assumed distribution of the response variable. Of course, this invariance does not hold for $[\boldsymbol{Z} \mid \boldsymbol{\mu}_Z, \psi]$.

The marginal likelihood, which depends on the observations $\boldsymbol{Z}$ and not on the unobserved random effects $\boldsymbol{u} \equiv (\boldsymbol{\eta}^\top, \boldsymbol{\xi}^\top)^\top$, is given by integrating out $\boldsymbol{u}$ from (13):

$$L^*(\boldsymbol{\theta}; \boldsymbol{Z}) \equiv \int_{\mathbb{R}^p} L(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u}) \mathrm{d}\boldsymbol{u}, \tag{14}$$

where p is the total number of random effects in the model. When the data are non-Gaussian, the integral in (14) is typically intractable and must be approximated either numerically or analytically. In **FRK** v2, we use a Laplace approximation, which we now describe.

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

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

where $\nabla_{\boldsymbol{u}}$ denotes the gradient with respect to $\boldsymbol{u}$. A second-order Taylor series approximation of $l(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u})$ about $\boldsymbol{u} = \hat{\boldsymbol{u}}$ results in an approximation of (13) that is Gaussian in terms of $\boldsymbol{u}$, with mean vector $\hat{\boldsymbol{u}}$ and covariance matrix $\boldsymbol{H}$. Substituting this approximation into (14) and evaluating the integral yields the Laplace approximation of the marginal likelihood, $L^*(\boldsymbol{\theta}; \boldsymbol{Z}) \approx L(\boldsymbol{\theta}; \boldsymbol{Z}, \hat{\boldsymbol{u}})(2\pi)^{\frac{p}{2}} |\boldsymbol{H}|^{\frac{1}{2}}$.

Note that $[\boldsymbol{u} \mid \boldsymbol{Z}, \boldsymbol{\theta}] \propto [\boldsymbol{u}, \boldsymbol{Z} \mid \boldsymbol{\theta}]$, and $[\boldsymbol{u}, \boldsymbol{Z} \mid \boldsymbol{\theta}]$ is equal to the complete-data likelihood function, $L(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u})$. Since the Laplace approximation replaces $L(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u})$ with a term that has the form of a Gaussian distribution, $\text{Gau}(\hat{\boldsymbol{u}}, \boldsymbol{H})$, in terms of $\boldsymbol{u}$, it follows that, approximately, $\boldsymbol{u} \mid \boldsymbol{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\boldsymbol{u}}, \boldsymbol{H})$. Further, **TMB** (Kristensen *et al.* 2016, see below) provides estimates of $\hat{\boldsymbol{u}}$ and $\boldsymbol{H}^{-1}$. This makes prediction of $\boldsymbol{u}$ and nonlinear functions of $\boldsymbol{u}$ via MC simulation straightforward; see Section 2.4.

Model fitting with ***TMB***

Given a C++ template function that defines $l(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u})$, **TMB** (Kristensen *et al.* 2016) computes the Laplace approximation of the log marginal likelihood, and automatically computes its derivatives. These quantities are then called from within **FRK** v2 by an optimising function specified by the user (`nlminb()` is used by default). **TMB** uses **CppAD** (Bell 2005) for automatic differentiation, and 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 very good computational efficiency. **TMB**'s implementation of automatic differentiation is a key reason why **FRK** v2 can cater for a variety of response distributions and link functions, as each combination does not need to be considered on a case-by-case basis.

Note that all parameters, fixed effects, and random effects, are treated as random in **TMB** (with a flat prior assumed if a prior is not provided). We fix the parameters and fixed effects to their posterior-mode estimates, and then treat them as non-random quantities.

2.4. Prediction and uncertainty quantification

We now discuss spatial prediction, and uncertainty quantification of the predictions. There are three possible quantities of interest in this framework: The latent process $Y(\cdot)$, the mean process $\mu(\cdot)$, and the noisy data process. Recall that the Laplace approximation implies that, approximately, $\boldsymbol{u} \mid \boldsymbol{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\boldsymbol{u}}, \boldsymbol{H})$; since $\boldsymbol{Y}$ is a linear function of $\boldsymbol{u}$, inference on $Y(\cdot)$ can hence be done using closed form solutions. However, the posterior distribution of non-linear functions of $Y(\cdot)$ (e.g., the mean process) are typically not available in closed form, and some type of approximation is required. We choose to use a MC framework.

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

$$\boldsymbol{Y}_{\text{MC}} \equiv \boldsymbol{T}\boldsymbol{A} + [\boldsymbol{S}\ \boldsymbol{I}]\,\boldsymbol{U}, \tag{15}$$

where each of the n_{MC} columns of the matrix $\boldsymbol{A}$ contain the estimated posterior mode of $\boldsymbol{\alpha}$, and each of the n_{MC} columns of the matrix $\boldsymbol{U}$ are draws from $\boldsymbol{u} \mid \boldsymbol{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\boldsymbol{u}}, \boldsymbol{H})$. We obtain MC samples of $\boldsymbol{\mu} \mid \boldsymbol{Z}, \boldsymbol{\theta}$ via $\boldsymbol{M} \equiv g^{-1}(\boldsymbol{Y}_{\text{MC}})$, where $g^{-1}(\cdot)$ is applied element-wise. Finally, MC samples of the noisy data process can be constructed straightforwardly using $\boldsymbol{M}$, since the distribution of an exponential family member depends only on its conditional mean (and a dispersion parameter, which we model as being constant throughout the spatial domain).

For each quantity, we use the posterior expectation as our predictor, which can be estimated by simply taking row-wise averages of the matrices 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 posterior predictive intervals. Hence, in **FRK** v2, we also use the MC sampling approach described above to compute user-specified percentiles of the posterior predictive distribution.

Arbitrary prediction regions

Often, one does not wish to predict over a single BAU, but over regions spanning multiple BAUs. Define the set of prediction regions as $D^P \equiv \{\tilde{B}_k : k = 1, \ldots, N_P\}$, where $\tilde{B}_k \equiv \cup_{i \in c_k} A_i$, and where $\tilde{c}_k$ is some non-empty set in the power set of $\{1, \ldots, N\}$. Like the observation supports, the prediction regions $\{\tilde{B}_k\}$ may overlap, and, in practice, may not include entire BAUs; our criteria for determining whether a prediction region contains a BAU is the same as that used for determining whether an observation support contains a BAU (see Section 2.2).

Prediction over D^P requires some form of aggregation across relevant BAUs. Since aggregation must be done on the response scale, we restrict prediction over arbitrary regions to the mean process (or the noisy data process). Consider $\boldsymbol{\mu}_P \equiv \{\mu(\tilde{B}_k) : k = 1, \ldots, N_P\}$, the mean process evaluated over the prediction regions. Just as $\boldsymbol{\mu}_Z$ was constructed from the BAU level mean process $\boldsymbol{\mu}$ via $\boldsymbol{C}_Z$, since each $\tilde{B}_k$ is a BAU or a union of BAUs, one can construct an $N_P \times N$ matrix

$$\boldsymbol{C}_P \equiv \left(\tilde{w}_{i,k}\mathbb{I}(i \in \tilde{c}_k) : i = 1, \ldots, N; k = 1, \ldots, N_P\right),$$

such that

$$\boldsymbol{\mu}_P = \boldsymbol{C}_P\boldsymbol{\mu}.$$

As before, the proportionality constants, $\{v_i : i = 1, \ldots, N\}$, for the weights $\{\tilde{w}_{i,k}\}$ are controlled by the `wts` field of the BAU object and the argument `normalise_wts`. For consistency between the model fitting and prediction stages, in **FRK** v2 we require that the same proportionality constants and the same value of `normalise_wts` are used in construction of both $\boldsymbol{C}_Z$ and $\boldsymbol{C}_P$.

MC samples of $\boldsymbol{\mu}_P \mid \boldsymbol{Z}, \boldsymbol{\theta}$ can be constructed via $\boldsymbol{M}_P \equiv \boldsymbol{C}_P\boldsymbol{M}$, where recall that $\boldsymbol{M}$ consists of MC samples of $\boldsymbol{\mu} \mid \boldsymbol{Z}, \boldsymbol{\theta}$. Predictions and uncertainty quantification of the predictions can then be computed straightforwardly.

2.5. Spatio-temporal framework

Extending **FRK** v2 to accommodate non-Gaussian spatio-temporal data involves using an additional dimension of basis functions. Let r_t and r_s denote the number of temporal and spatial basis functions, respectively. Denote $\boldsymbol{Q}_t$ and $\boldsymbol{Q}_s$ as the precision matrices of the random coefficients associated with the temporal basis functions and spatial basis functions, respectively. We model the $r_t r_s \times r_t r_s$ precision matrix of the $r_t r_s$ spatio-temporal random coefficients associated with basis functions created through the tensor product of the r_t temporal and r_s spatial basis functions as

$$\boldsymbol{Q} = \boldsymbol{Q}_t \otimes \boldsymbol{Q}_s.$$

The assumption of separability between time and space, and hence the ability to write $\boldsymbol{Q}$ as a Kronecker product, leads to significant computational savings. **FRK** v2 uses an AR1 model for the random coefficients associated with the temporal basis functions.

In a spatio-temporal setting, it is possible that each spatial BAU is observed multiple times. Furthermore, it is also possible that the fine-scale variation is not constant over the spatial domain D. In these situations, a better fit may be obtained by allowing each spatial BAU to be associated with its own fine-scale variance parameter. Let N_s and N_t denote the number of spatial and temporal BAUs, respectively (so that $N = N_s N_t$). Then, instead of modelling $\boldsymbol{\xi} \sim \text{Gau}(\boldsymbol{0}, \sigma^2_\xi \boldsymbol{I})$, **FRK** v2 also allows one to model $\boldsymbol{\xi} \sim \text{Gau}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\xi\right)$, where

$$\boldsymbol{\Sigma}_\xi \equiv \begin{pmatrix} \text{diag}(\boldsymbol{\sigma}^2_\xi) & & \\ & \ddots & \\ & & \text{diag}(\boldsymbol{\sigma}^2_\xi) \end{pmatrix}, \tag{16}$$

$\boldsymbol{\sigma}^2_\xi \equiv (\sigma^2_{\xi,1}, \ldots, \sigma^2_{\xi,N_s})^\top$, and the BAUs are assumed to be ordered such that space runs faster than time. This model for $\boldsymbol{\Sigma}_\xi$ is flagged by setting `fs_by_spatial_BAU = TRUE` in the `SRE()`

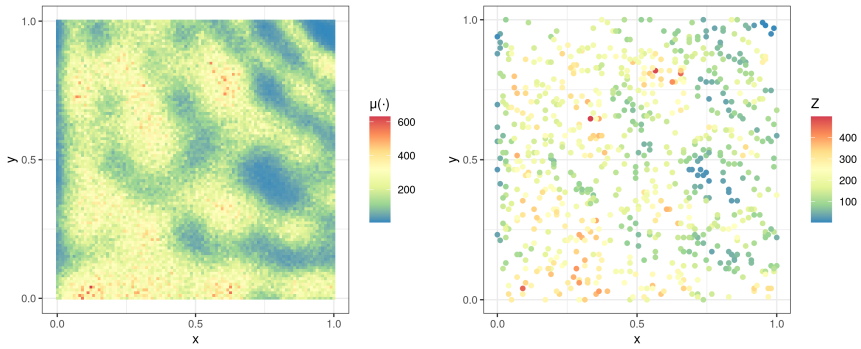

Figure 2: Simulated point-referenced spatial Poisson data set, and the true field from which the data was generated, used for the example presented in Section 3.1. (Left Panel) True mean process $\mu(\cdot)$. (Right panel) Simulated observations.

function, and is particularly useful when the number of spatial BAUs (and hence variance parameters to estimate) is relatively low and we have observed each spatial BAU over many time-points; see, for instance, the example presented in Section 4.4.

3. Usage of new features

We now demonstrate the new features in **FRK** v2, an overview of which is presented in Table 1. The primary new feature in **FRK** v2 is the packages ability to cater for non-Gaussian data models: A full list of available data models and link functions is shown in Table 2. In Sections 3.1 and 3.2, we illustrate use of **FRK** v2 using non-Gaussian spatial point-referenced and area-referenced data, respectively. Finally, in Section 3.4, we show the potential improvement in predictive performance of **FRK** v2 over **FRK** v1 when the data are Gaussian, thanks to the support for an increased number of basis functions in **FRK** v2. For all results presented in the remainder of this paper, reproducible code is provided at `https://github.com/MattSainsbury-Dale/FRKv2_src`.

3.1. Example: Non-Gaussian, point-referenced spatial data

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. The true mean process over D and the data are shown in Figure 2.

The first step when using **FRK** v1/v2 is to create basis functions and BAUs, which can be done automatically using the helper functions `auto_BAUs()` and `auto_basis()`; see Zammit-Mangion and Cressie (2021) for details. Next, an '`SRE`' object is initialised using `SRE()`, within which we specify the data model, the link function, and the parameterisation of $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$. We fit the model using `SRE.fit()`. These steps may be performed in a single line of code with the convenient wrapper function `FRK()`. Note that when the data are non-Gaussian or a non-identity link function is chosen, `FRK()` automatically selects `K_type = "precision"`.

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

Table 1: Important extensions to function arguments in **FRK** v2.

Function	Argument	Use
`SRE()`	`response`	A string indicating the data model.
	`link`	A string indicating the link function.
	`K_type`	A string indicating the parameterisation of $\text{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$; the newly permissible value, `"precision"`, specifies that a sparse precision matrix should be used.
	`normalise_wts`	A flag controlling whether weights in $\boldsymbol{C}_Z$ and $\boldsymbol{C}_P$ should be normalised, so that aggregation of the mean corresponds either to a weighted sum or a weighted average.
	`fs_by_spatial_BAU`	A flag controlling whether each spatial BAU is given its own fine-scale variance parameter; only applicable in a spatio-temporal setting.
`SRE.fit()`	`method`	A string indicating the method of model fitting; the newly permissible value, `"TMB"`, is required whenever a non-Gaussian data model or non-identity link function is used.
	`known_sigma2fs`	Allows one to fix the fine-scale variance to a known value.
`predict()`	`type`	A vector of strings indicating the quantities of interest for which inference is desired. The inclusion of `"link"`, `"mean"`, and `"response"` in `type` respectively indicate that inference on $Y(\cdot)$, $\mu(\cdot)$, or the noisy data process is desired.
	`percentiles`	Numeric vector indicating the percentiles of the posterior predictive distribution(s) to be computed.
	`n_MC`	Integer indicating the number of MC samples at each BAU.
`auto_BAUs()`	`spatial_BAUs`	The spatial BAUs in a spatio-temporal setting. If `NULL`, the spatial BAUs are constructed automatically.
`plot()`	-	A method for visualising the data, predictions, and uncertainty quantification of the predictions given an '`SRE`' object and the result of a call to `predict()` on the '`SRE`' object.

Table 2: Combinations of exponential family member response distributions and link functions available in **FRK** v2. A '✓' indicates a combination is supported. A '•' indicates a combination is allowed, however, due to the implied range of μ, the support of the observations, and the form of probability density function of that family, nonsensical results are possible; if one of these problematic combinations is chosen, a warning is given to the user. Finally, blank entries indicate that the combination is not allowed.

		Link Function				
		identity	inverse	log	square-root	logit/probit/cloglog
Family	Gaussian	✓	✓	•	•	
	Poisson	•	•	✓	✓	
	gamma	•	•	✓	✓	
	inverse-Gaussian	•	•	✓	✓	
	negative-binomial			✓	✓	✓
	binomial					✓

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 process $Y(\cdot)$ and the mean process $\mu(\cdot)$. The `percentiles` argument allows the computation of percentiles of the posterior predictive distributions, and hence posterior predictive intervals; by default, the 5th and 95th percentiles are computed.

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

When `method = "TMB"`, the returned object 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, we can generate a list of '`ggplot`' (Wickham 2016) objects of the predictions and associated uncertainty using the function `plot()`.

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

The '`ggplot`' objects can then easily be arranged in a grid using various dedicated packages; we used **ggpubr** (Kassambara 2020). Figure 3 shows prediction and uncertainty quantification of the predictions for the latent process $Y(\cdot)$ and the mean process $\mu(\cdot)$. The predictions of $\mu(\cdot)$ are reasonable given the data and the true process shown in Figure 2. The prediction uncertainty for $Y(\cdot)$ is relatively flat, with uncertainty increasing in regions of data paucity; on the other hand, the prediction uncertainty for $\mu(\cdot)$ is roughly proportional to its prediction. The 'bullseye' points of low uncertainty, visible for both processes, correspond to the observed locations. The point-like nature of this reduction in uncertainty arises from the fine-scale

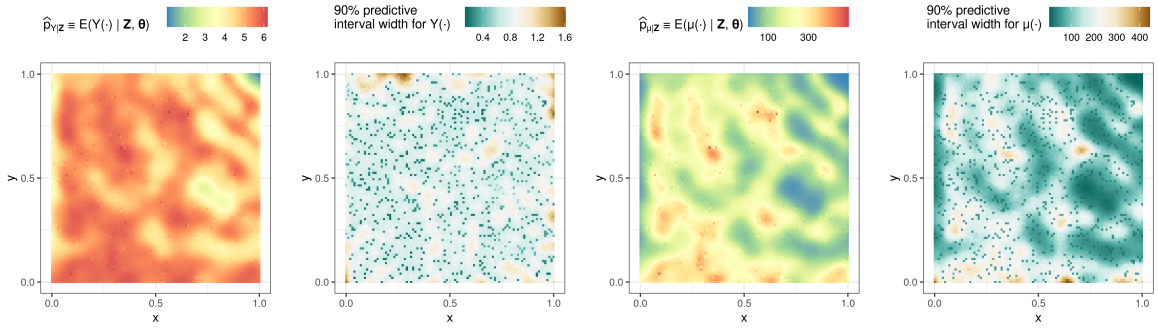

Figure 3: The prediction and prediction uncertainty quantification using the data shown in Figure 2. (Left panel) Prediction of the latent process, $Y(\cdot)$. (Centre-left panel) Width of the 90% central predictive interval of the latent process. (Centre-right panel) Prediction of the mean process, $\mu(\cdot)$. (Right panel) Width of the 90% central predictive interval of the mean process.

random effects, $\boldsymbol{\xi}$, being modelled as mutually independent at the BAU level: unobserved BAUs do not borrow strength from the inferred fine-scale random effect at neighbouring observed BAUs.

3.2. Example: Non-Gaussian, areally-referenced spatial data and change of support

In this section, we illustrate **FRK** v2 on simulated negative-binomial, areally-referenced, spatial data, as well as its use in predicting over large areas.

In the the first step of data simualtion, we define two square grids: The first is at a fine resolution and corresponds to the BAUs, and the second is at a coarser resolution and corresponds to areal data supports. To get a handle on the fine-scale variation parameter, we use some of the BAUs as data supports: Hence we have a mixture of coarse- and fine-scale observations. We define the probability process evaluated over the BAUs by passing a sum of trigonometric functions through the logistic function. We then construct the mean process evaluated over the BAUs; as we are simulating negative-binomial data, this requires specification of a size parameter, and for simplicity we use $k = 50$ for each BAU. We then sum the mean process over the data supports, and simulate data using the aggregated mean process. Finally, we exclude some observations to form a training set. This simulation procedure is illustrated in Figure 4.

Now we construct and fit the '`SRE`' object using `FRK()`. By setting `normalise_wts = FALSE`, we indicate that the weights of $\boldsymbol{C}_Z$ and $\boldsymbol{C}_P$ should not be normalised, so that the aggregation of the mean process from the BAU level to the data support level corresponds to a sum. For binomial and negative-binomial data, the size parameter must be provided: In general, we need the size parameter of every observed BAU. When each observation is associated with exactly one BAU (e.g., point-referenced data, or areal data where the BAUs and observation supports coincide), the user can provide the size parameter with the observations; when some observations are associated with multiple BAUs, the user must provide the size parameter at the BAU level (for all observed BAUs).

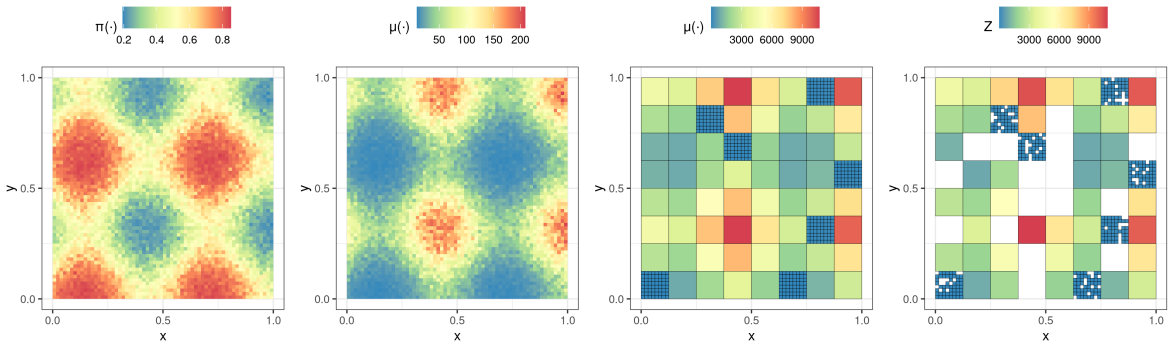

Figure 4: Simulated, areal negative-binomial data set used in the illustrative example of Section 3.2. (Left panel) True probability process evaluated over the BAUs. (Centre-left panel) True mean process evaluated over the BAUs. (Centre-right panel) True mean process aggregated to the data support level. (Right panel) Simulated data at the data support level, with some observations omitted; this is the data used for model fitting.

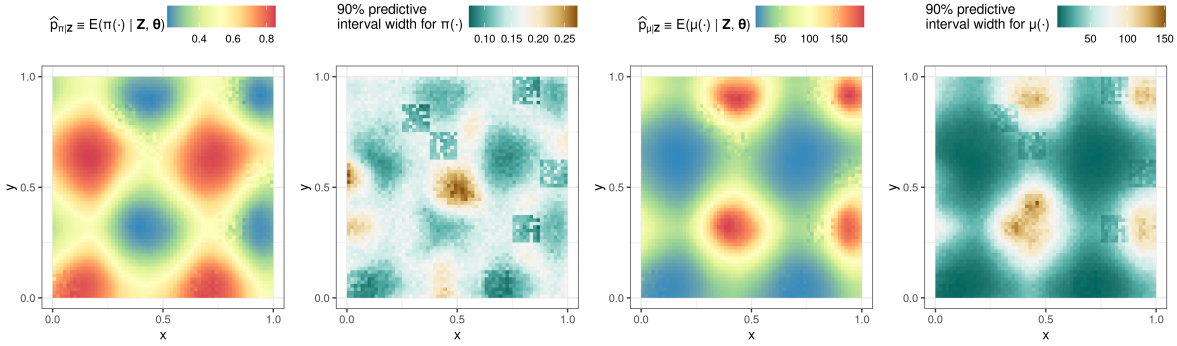

Figure 5: Prediction and uncertainty quantification of the predictions for the simulated negative-binomial areal example. (Left panel) Prediction of the probability process $\pi(\cdot)$. (Centre-left panel) Width of the 90% central predictive interval of the probability process. (Centre-right panel) Prediction of the mean process, $\mu(\cdot)$. (Right panel) Width of the 90% central predictive interval of the mean process.

```
R> BAUs$k_BAU <- 50
R> S <- FRK(f = Z ~ 1, data = list(zdf), BAUs = BAUs,
+     response = "negative-binomial", link = "logit", normalise_wts = FALSE)
```

Next, we predict over the BAUs.

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

Figure 5 shows the predictions and uncertainty quantification of the predictions for both the mean process, $\mu(\cdot)$, and the probability process, $\pi(\cdot)$, at the BAU level. We observe agreement between the fields shown in Figure 4 and the corresponding predictions. The

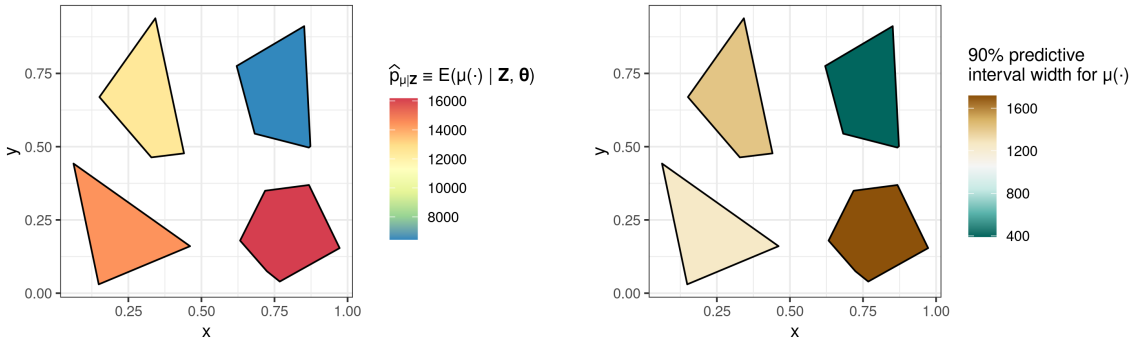

Figure 6: Prediction (left panel) and prediction uncertainty (right) of the mean process, $\mu(\cdot)$, when predicting over arbitrary polygons in the toy example of Section 3.2. Note that these polygons have equal area.

prediction uncertainty of $\mu(\cdot)$ is roughly proportional to its prediction. In contrast, the prediction uncertainty of $\pi(\cdot)$ is low when the prediction is near 0 or 1, and increases when the prediction is near 0.5: This is expected from properties of the negative-binomial distribution. Uncertainty in both quantities is lower over areas in which we have fine-scale data. The mean empirical coverage from the 90% posterior predictive intervals was 90.9%, which is almost nominal, and very reasonable given the difficulty inherent to spatial change-of-support problems.

To emphasise that the prediction polygons are unrelated to the BAUs and data supports, we demonstrate prediction over a handful of irregularly-shaped areas (defined as a '`SpatialPolygons*`' object).

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

Recall from Section 2.4 that we restrict prediction to $\mu(\cdot)$ over arbitrary polygons. Figure 6 shows the predictions and uncertainty quantification of the predictions over the irregularly-shaped areas.

3.3. Spatio-temporal extensions

FRK v2 now also caters for non-Gaussian, point- and areally-referenced, *spatio-temporal* data. For the sake of brevity, we will not use a simple example to show this, and instead refer readers to the application study of Section 4.4.

3.4. Support for increased number of basis functions

The efficiency of **TMB** and our use of sparse precision matrices means that **FRK** v2 is now better equipped than **FRK** v1 to use a large number of basis functions. The predictive performance of the framework can be closely related to the number of basis functions, as shown in the following examples. Appendix C defines the scoring rules used throughout the remainder of this paper.

We repeated the analysis in Section 3.1 using one, two, and three resolutions of basis functions; Table 3 shows the results for each run. Clearly predictive performance improves as the number

Table 3: Diagnostics comparing the predictive performance when using a range of basis function resolutions and with point-referenced count data. The diagnostics are the root mean-square prediction error (RMPSE), the continuous ranked probability score (CRPS), and the empirical coverage (Cvg90) and interval score (IS90) resulting from a central prediction interval with a nominal coverage of 90%. The diagnostics are in regards to prediction of the true mean process, $\mu(\cdot)$, and are averaged over all unobserved locations.

Resolutions (basis functions)	RMSPE	CRPS	Cvg90	IS90	Run Time (Min.)
1 (9)	83.15	47.63	0.896	377.26	0.067
2 (90)	53.96	28.54	0.893	224.24	0.128
3 (819)	47.12	24.31	0.895	182.59	0.521

Table 4: Numerical scoring for each competing method on the MODIS data, as presented in Heaton *et al.* (2019).

Method	MAE	RMSPE	CRPS	IS95	Cvg95	Run Time (Min.)	Cores Used
FRK v2	1.30	1.69	0.92	8.32	0.93	72.27[a]	1[b]
FRK v1	1.96	2.44	1.44	14.08	0.79	2.32	1
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

[a] **FRK** v2 was implemented in a different computing environment than the other models, and so 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 Becker computing environment (256 GB of RAM and 2 Intel Xeon E5-2680 v4 2.40GHz CPUs with 14 cores each and 2 threads per core - totaling 56 possible threads for use in parallel computing) located at Brigham Young University (Heaton *et al.* 2019).
[b] **TMB** supports the use of multiple cores, but this is not yet implemented in **FRK** v2.

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. This important property is in large part due to the fine-scale random variation term, $\xi(\cdot)$, in (1).

Although the primary motivation for **FRK** v2 is the provision of non-Gaussian data models, the ability to use a large number of basis functions also leads to better predictions in a Gaussian setting. We show this through the comparative study provided in Heaton *et al.* (2019). The data used in that study was made up of a training and a test set consisting of 105,569

and 42,740 observations, respectively; see Heaton *et al.* (2019) for a detailed description of the data. Table 4 replicates Table 3 of Heaton *et al.* (2019), with an additional entry corresponding to **FRK** v2, wherein many more basis functions are used than was practical with **FRK** v1. Specifically, **FRK** v1 used 485 basis functions, whilst **FRK** v2. uses 12114. The results show that the increased number of basis functions significantly improves the diagnostic scores of **FRK**, and the result are now comparable to those of MRA. To achieve these improvements over **FRK** v1, we only had to specify `nres = 4`, `K_type = "precision"` and `method = "TMB"` in `auto_basis()`, `SRE()`, and `SRE.fit()`, respectively; the rest of the **FRK** v1 code as used in the competition was left unchanged.

4. Application and comparison studies

We now provide several application case studies using **FRK** v2. In Section 4.1, we present a comparison study between **FRK** v2 and other packages which cater for non-Gaussian data models. In Section 4.2, we demonstrate block prediction using contaminated soil data, and compare our results to other modelling approaches. 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. In Section 4.4, we provide a non-Gaussian spatio-temporal example through modelling crime counts in the city of Chicago over the first two decades of the 21st century.

4.1. Comparative study: MODIS cloud data

In this section we compare out-of-sample predictions from **FRK** v2 to those from the R packages **INLA** (Lindgren and Rue 2015), **spNNGP** (Finley *et al.* 2020), **spBayes** (Finley *et al.* 2015), and **mgcv** (Wood 2017) using a 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). Data collected from the MODIS instrument have been used in several related works; see, for instance, Sengupta and Cressie (2016) and Zammit-Mangion, Ng, Vu, and Filippone (2021). For this comparative study, data pre-processing involved first coarsening the image from over 10 million pixels to a more manageable 33750 pixels, by creating a 150×225 grid and computing the mean value of the response within each grid cell. Then, as the data provided by the MODIS instrument is continuous (measuring radiance in units of $\mathrm{W/m^2/\mu m/st}$), we applied a reasonable threshold to obtain a binary version of the data (i.e., cloud, or no-cloud).

We considered two types of sampling schemes for model testing. The first was missing-at-random (MR), whereby we randomly selected a sub-sample of pixels to act as training data. Under the MR sampling scheme, we randomly sampled 6000 pixels for training, leaving 27750 pixels for testing. The second sampling scheme, which we refer to as 'missing-in-a-block' (MB), involved excluding all pixels within a block for training, and using pixels inside the block for testing. The block is a 30×30 square (900 pixels) in the middle of the spatial domain of interest. The training and test sets under the two sampling schemes are shown in Figure 7.

The software used in this study 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 a pre-processing model-selection phase by splitting the training set in two, and then

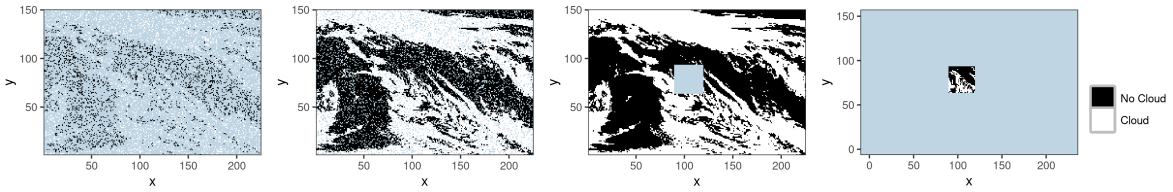

Figure 7: MODIS data used in the comparative study of Section 4.1; a blue background is used to make the 'No Cloud' and the 'Cloud' pixels easier to distinguish. (Left panel) The missing-at-random sample used for training. (Centre-left panel) The missing-at-random test data. (Centre-right panel) The 'missing-in-a-block' sample used for training. (Right panel) The 'missing-in-a-block' test data.

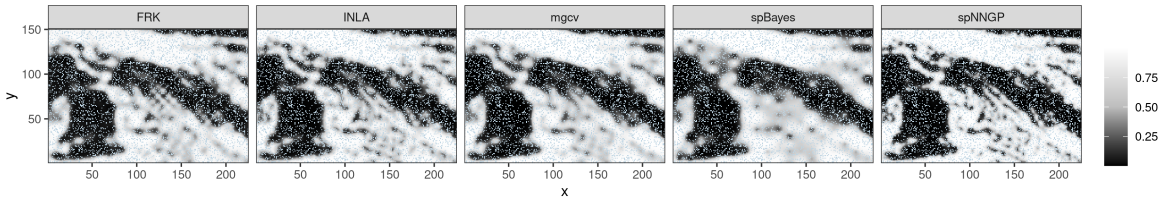

Figure 8: Predictions of the probability of cloud resulting from the missing-at-random data shown in Figure 7. Note that the training locations are indicated by blue pixels.

using one half for model fitting and the other half for model validation. In this way, we were able to test a large number of arguments for each package, and choose the best combination in terms of predictive performance and run time. For the methods requiring specification of a link function, we used the standard logit link function. For **FRK** v2, we used four resolutions of basis functions; a total of 11,130 basis functions. For **INLA**, we discretised the domain into 8671 elements. We used the `bam()` function from **mgcv**, which is similar to generalised additive model function `gam()`, but optimised for large data sets, with 3000 knots. For **spBayes**, we used 400 knots; increasing the number of knots further was computationally prohibitive. We found that the default option of considering 15 neighbours at a time when using **spNNGP** was appropriate. The packages **spNNGP** and **spBayes** use Markov chain Monte Carlo (MCMC): At both training and test locations, we used 10000 total MCMC samples, a burn-in of 6000, and a thinning factor of 10; hence, 400 approximately independent samples from the predictive distribution of the process were available at each spatial location. The number of cores used for **spNNGP** can be controlled through the argument `n.omp.threads`. Setting `n.omp.threads` to be greater than 1 did not work on our computing system (a known issue documented in the **spNNGP** package manual); hence, our reported run-times for **spNNGP** are for a single core.

For each method and each sampling scheme, we predicted the probability of cloud at each pixel. Figure 8 shows the predictions resulting from the MR data shown in Figure 7. The predictions of **FRK** v2, **INLA**, and **spNNGP** are similar, while the predictions of **mgcv** are slightly smoother than those from the aforementioned packages. The predictions of **spBayes**

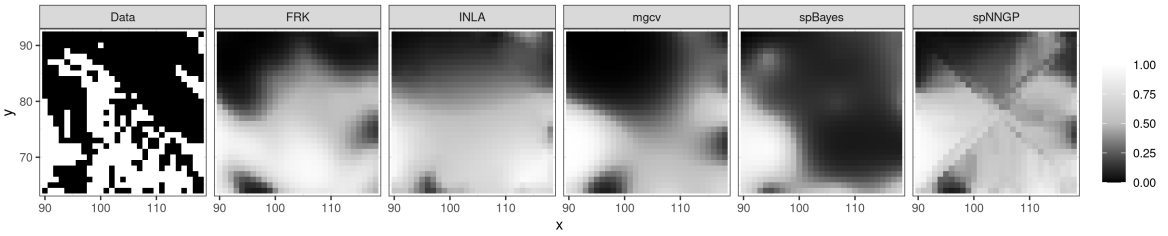

Figure 9: Predictions of the probability of cloud resulting from the 'missing-in-a-block' data shown in Figure 7. Here, we have shown only the testing locations, which corresponds to the 30×30 block near the centre of the spatial domain; the test data corresponding to this block is shown in the left-most panel of this figure.

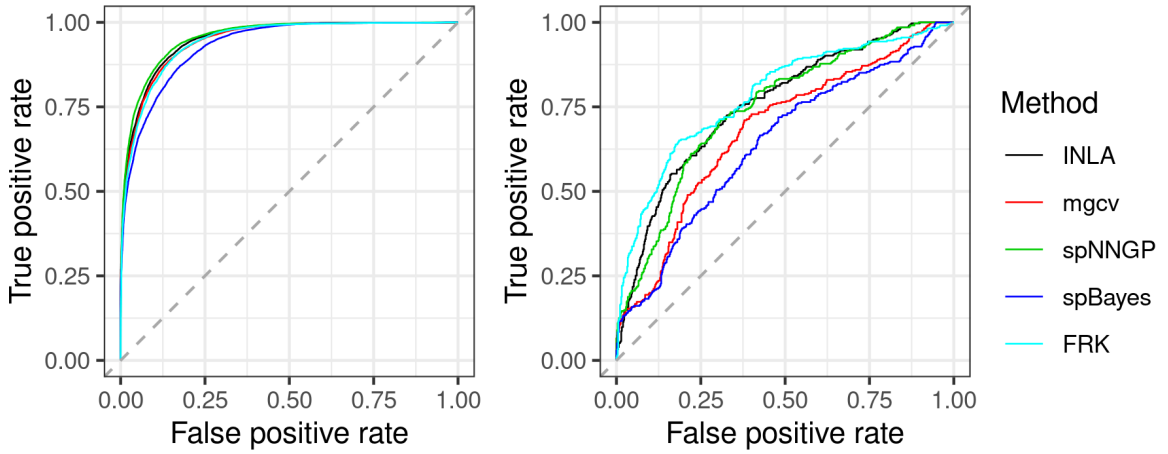

Figure 10: ROC curves for the training/test sets displayed in Figure 7. (Left panel) ROC curves generated from the 'missing-at-random' data. Note that there is some degree of overlap between **FRK**, **INLA**, and **spNNGP**. (Right panel) ROC curves generated from the 'missing-in-a-block' data.

are even smoother; this is due to the small number of knots. Figure 9 shows the predictions resulting from the MB data shown in Figure 7. **FRK** 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.

The packages used in this study can provide prediction standard errors associated with the probability process. However, the underlying distribution of the probability process is unidentifiable, as the posterior predictive distribution, $Z^* \mid \boldsymbol{Z}$, for some validation datum Z^*, depends only on the posterior expectation of the probability parameter at the corresponding location, $\mathbb{E}(\pi^* \mid \boldsymbol{Z})$. For this reason, we do not attempt to validate the prediction intervals, and instead focus our efforts on predictive accuracy.

Table 5: Diagnostic results for the MODIS comparison study. Best performers for a given diagnostic are boldfaced.

Scheme	Method	Brier score	AUC	Run Time (Min.)
MR	FRK	0.087	0.955	**5.47**
	INLA	0.090	0.951	54.44
	mgcv	0.091	0.948	45.77
	spBayes	0.105	0.932	68.01
	spNNGP	**0.083**	**0.956**	12.35
MB	FRK	**0.192**	**0.769**	**9.29**
	INLA	0.200	0.757	141.85
	mgcv	0.219	0.701	186.67
	spBayes	0.247	0.632	489.41
	spNNGP	0.198	0.749	59.30

To assess predictive accuracy, we compared the predictions from all models in terms of the Brier score (Gneiting, Balabdaoui, and Raftery 2007, Sec. 3), and the area under the receiver operating characteristic (ROC) curve (AUC). The Brier score assesses how close the predicted probability of cloud is to the truth; it is a negatively oriented rule, where low scores indicate accurate predictions of the probability of cloud. In contrast, higher AUC scores are preferred. The results for each method and each sampling scheme are reported in Table 5; the ROC curves are shown in Figure 10. For the MR scheme, there is little discernible difference between **FRK**, **INLA**, **mgcv**, and **spNNGP**. However, as one may expect upon viewing the predictions in Figure 8, **spBayes** performs poorly in comparison to the other packages due to the small number of knots. The task of prediction over a completely unobserved region is challenging, and so it is no surprise that the diagnostics for the MB scheme are significantly worse than the MR scheme. In this case, we see **FRK**, **INLA**, and **spNNGP** performing slightly better than **mgcv**, which in turn performs better than **spBayes**. Note that all run times increased under the MB scheme; however, **FRK** v2 increased by the smallest amount (increasing by a factor of less than 2), while the run times for other packages increased by between a factor of 3 and 10. Given that the training sample size is significantly larger in the MB scheme than under the MR scheme (32,850 pixels compared to 6000 pixels), this suggests that **FRK** v2 is well suited to fitting and predicting with large sample sizes. Overall, these results suggest that **FRK** v2 is comparable to other packages in this application. The advantages of **FRK** v2 lie in the ease with which it allows one to do other more elaborate analyses with non-Gaussian data, as shown in the next sections.

4.2. Block prediction: Contaminated soil

Between 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 we use in this example contains Am concentrations (in 10^3 counts per minute) in a spatial domain immediately surrounding Ground Zero (GZ), the location where the devices were detonated, and was previously analysed by Huang, Yao, Cressie, and Hsing (2009) and Paul and Cressie (2011). The total number of measurements (including some that are collocated) is 212. The left and centre panels of Figure 11 shows the data on the original scale and on the log scale, respectively. Paul and Cressie (2011) note that the Am

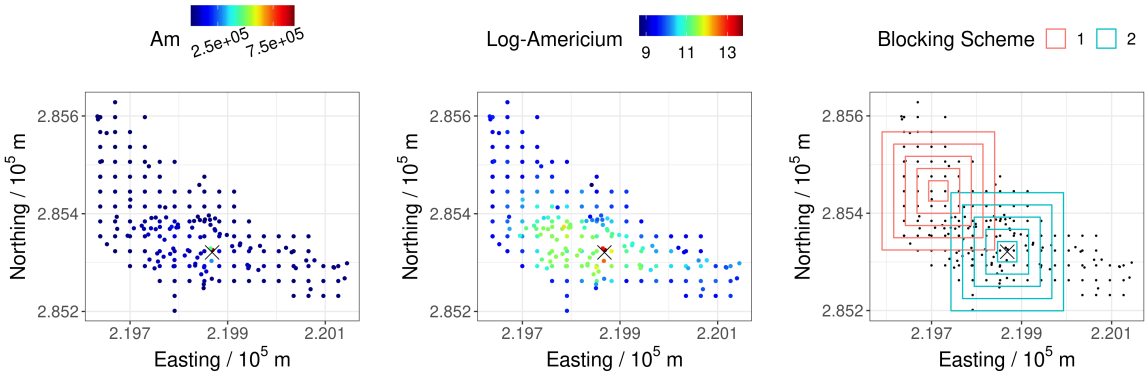

Figure 11: Americium soil data. The '×' denotes Ground Zero (GZ), where the devices were detonated. (Left panel) Am concentrations on the original scale. (Centre panel) Am concentrations on the log scale. (Right panel) Americium soil data blocking schemes: Scheme 1 (red), centred away from GZ, and Scheme 2 (blue), centred on GZ.

concentrations are clearly lognormally distributed, and that soil remediation is often made 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 11 shows two blocking schemes which we will predict over: Both schemes contain 5 blocks, but one scheme is centred on GZ, and the other is centred away from GZ.

As in Paul and Cressie (2011), we use a piecewise linear trend, where observations within a distance of 30.48m from GZ follow a different trend to those observations beyond 30.48m from GZ. The following code constructs the required BAU level covariates.

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

Modelling for this problem is done by setting `response = "Gaussian"` and `link = "log"`. In order to mimic lognormal block kriging, here we fix the measurement error standard deviation to a small value prior to model fitting.

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

By predicting over the BAUs, one may generate predictions over the entire spatial domain. Alternatively, by passing a '`SpatialPolygonsDataFrame`' object into the `newdata` argument of `predict()`, one may straightforwardly generate block-level predictions.

To compare our predictions, we used the R package **georob** (Papritz 2020), which implements an approximately unbiased back-transformation of kriging predictions of log-transformed data (Cressie 2006). Kriging does not scale well for large sample sizes, however the size of this data set is small. The package **georob** provides users with two approaches to lognormal

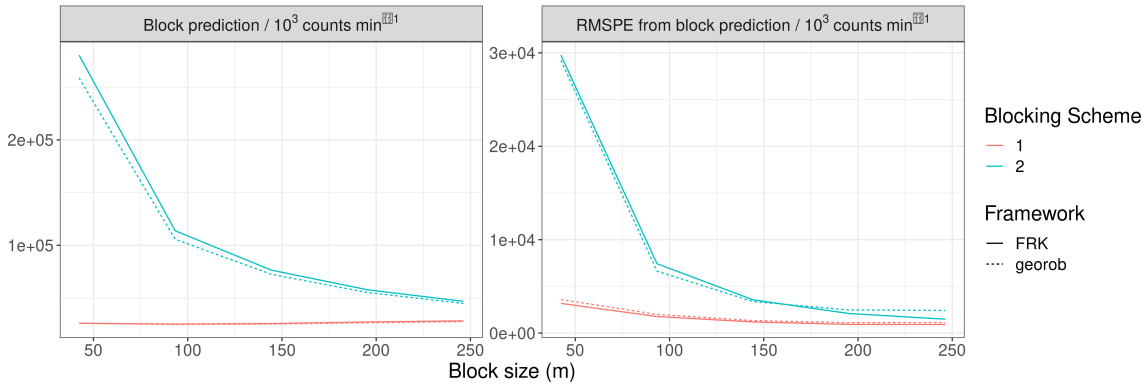

Figure 12: Predictions and root mean squared prediction error (RMSPE) against block size for the two blocking schemes. (Left panel) Block-predictions of Am concentrations against block size $|B|^{1/2}$. (Right panel) RMSPE of block predictions of Am concentrations against block size $|B|^{1/2}$. In both plots, the red line corresponds to Scheme 1 and the blue line corresponds to Scheme 2.

block kriging; we used the 'optimal predictor', as recommended by the **georob** manual when predicting over large blocks. Figure 12 shows the block predictions and associated RMSPE obtained using **FRK** v2 and **georob** for the two blocking schemes shown in Figure 11. The similarity in results lends confidence that the predictions and associated prediction standard errors obtained using **FRK** v2 are reasonable.

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 New South Wales containing 7909 SA1 regions, 180 SA2 regions, and 31 SA3 regions, and aim to infer 'poverty' levels at the SA1 and SA3 regions, just from a data set containing mostly SA2 data and a small amount of SA1 data. The data was collected in the Census of 2011, and it consists of the number of families of various types within a range of weekly income brackets; we provide further details, and the way in which we define the poverty line for each family type, in Appendix D. Note that data at the SA1 and the SA3 regions are available, and we use these to validate our down-scaled and up-scaled predictions.

It is often the case that sampling once from a large area is relatively inexpensive compared to acquiring multiple samples from small areas. Our training data, shown in Figure 13, is reflective of such a scenario. It includes mostly SA2 regions, but some SA1 regions have also been included.

In this example, we use the SA2 (and some SA1) region data for training the model, and the SA1 regions as the BAUs; these are passed as '`SpatialPolygonsDataFrame`' objects to `FRK()`. We also set `normalise_wts = FALSE`, which indicates that we wish to model the mean process in a given data polygon as the sum (rather than the average) of the mean process

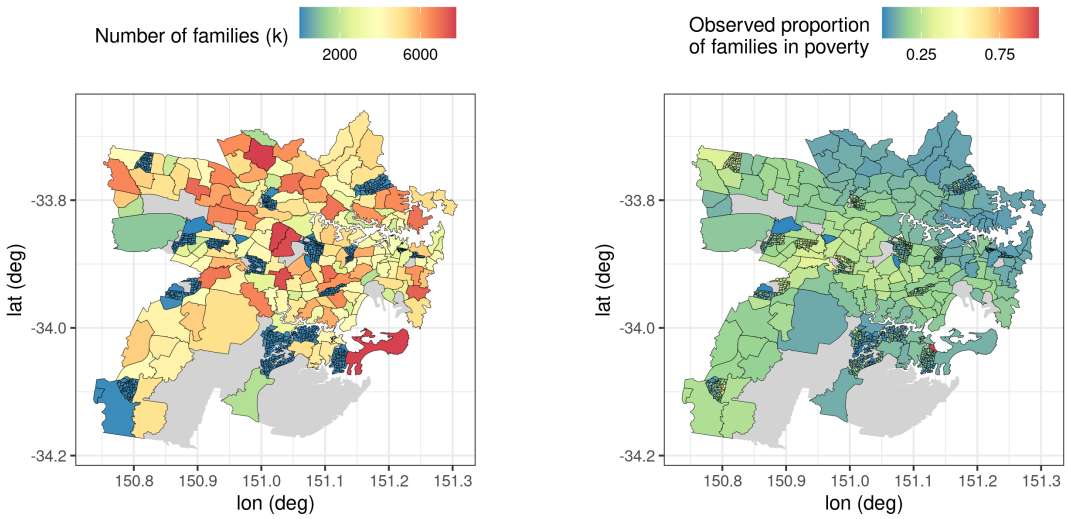

Figure 13: Training data used for modelling the number (or proportion) of families 'in poverty' (see main text for how we define 'in poverty'). (Left panel) The total number of families. (Right panel) The observed proportion of families in poverty, computed by dividing the number of families in poverty by the total number of families. Grey regions correspond to SA regions in which the total number of families is zero.

over the SA1s.

```
R> S <- FRK(f = total_poverty_count ~ 1,
+     data = list(SA2_and_some_SA1s), BAUs = SA1s,
+     response = "binomial", link = "logit", normalise_wts = FALSE)
```

Now we predict over the SA1 regions.

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

Since the SA1 regions are the BAUs, we can predict both the probability and mean processes over the SA1 regions: We focus on the probability process, which is independent of the size parameter. The predictions and associated uncertainty over the SA1 regions are shown in Figure 14, which was generated using `plot()`.

Predicting over different spatial supports is straightforward with **FRK** v2. To predict over the SA3 regions, we simply set `newdata` to a '`SpatialPolygonsDataFrame`' object containing the SA3 regions.

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

Figure 15 shows the SA3 region predictions and associated uncertainty quantification: Since the SA3 regions are not the BAUs, this time we are restricted to prediction of the mean process. Again, this graphic was generated using `plot()`.

We assessed the models ability quantify uncertainty over the SA1 regions by computing the empirical coverage from 90% prediction intervals. The empirical coverage was 90.8%, which

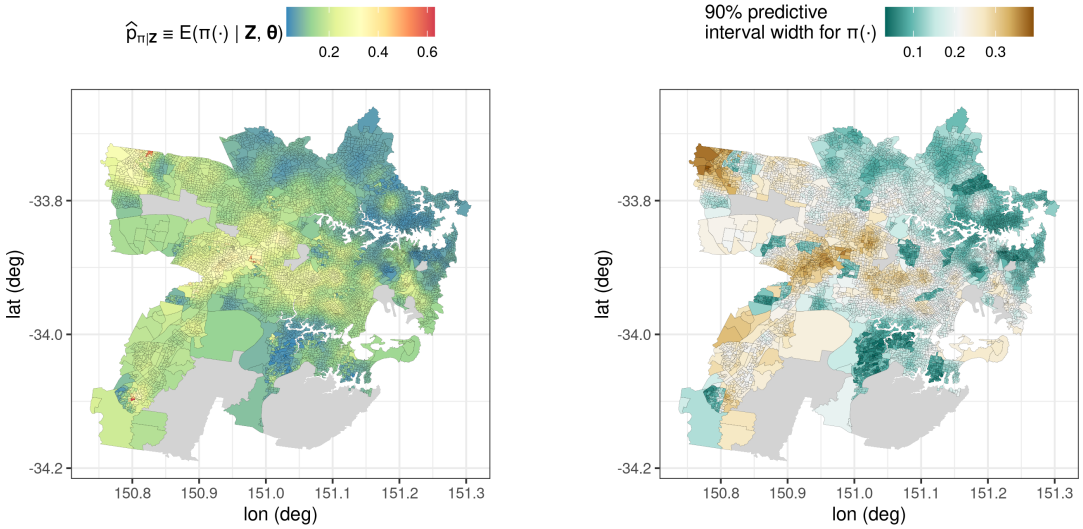

Figure 14: SA1 level predictions. (Left panel) Prediction of the probability process, $\pi(\cdot)$, representing the proportion of families in poverty, over the SA1 regions. (Right panel) 90% prediction interval width of the probability process. Grey regions correspond to SA2 regions in which the total number of families is zero, and hence are omitted from the study.

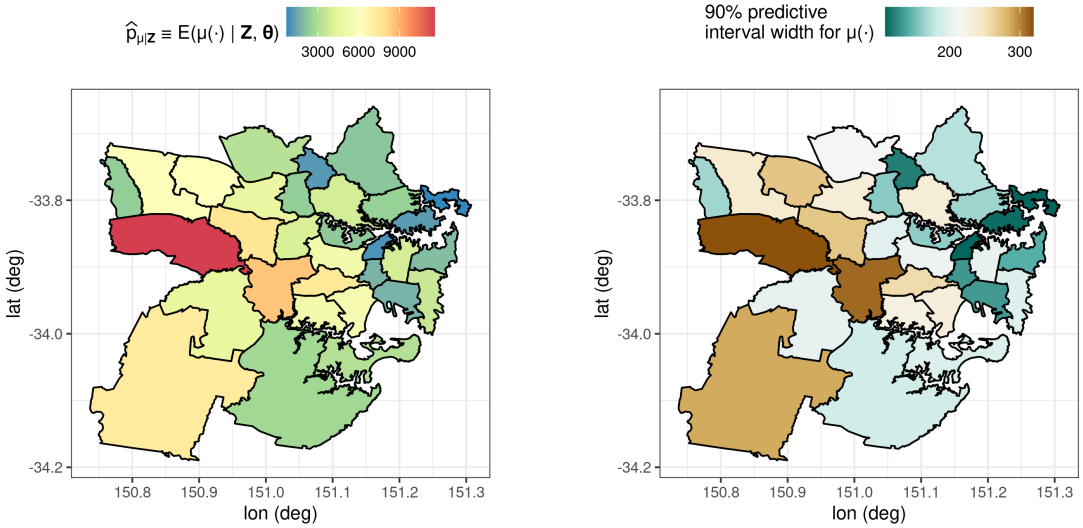

Figure 15: SA3 level predictions. (Left panel) Prediction of the mean process, $\mu(\cdot)$, representing the expected number of families in poverty, over the SA3 regions. (Right panel) The 90% prediction interval width of the mean process.

is almost nominal. The inclusion of some fine-scale data (SA1 region data) greatly aids in the estimation of the fine-scale variance parameter, σ^2_ξ. If only coarse-resolution data are available (i.e., all data supports are associated with multiple BAUs), in order to avoid identifiability

issues, **FRK** v2 fixes σ^2_ξ prior to model fitting with **TMB**. In this situation, if σ^2_ξ is unknown, **FRK** v2 generates a rough and possibly unreliable estimate. If one does know σ^2_ξ, or can obtain a reliable estimate of it (for example, using past census data), one may specify it using the argument `known_sigma2fs`.

We conclude this example by noting that, although the prediction polygons, data supports, and BAUs in this study have a nested relationship, in general these elements can be entirely unrelated: Prediction in **FRK** can be done over any arbitrary, user-specified polygons. See Section 3.2 for an illustration.

4.4. Non-Gaussian 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, their boundaries having changed little since their inception in the 1920's (The University of Chicago Library 2020). In this study, we model the number of crimes in each CA between the years 2001 and 2019. A full list of crimes committed in Chicago during this period is provided by the Chicago Police Department, and is available for download from the open data source website Plenario (Urban Center for Computation and Data and University of Chicago 2020). We considered crimes labelled as assault or battery; roughly 1.75 million crimes in total. Note that **FRK** bins data falling into the same BAU, so the final number of observations post-binning is significantly less. 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.

In this example, we use the CAs as our spatial BAUs. This can be done straightforwardly by reading in the shapefile of the CAs as a '`SpatialPolygonsDataFrame`' object. Spatio-temporal BAUs may then be constructed by passing the CAs and data into `auto_BAUs()`.

```
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 Combined Community Data Snapshots provided by the Chicago Metropolitan Agency for Planning (2017). It is difficult to obtain population data for every year, so we assume that population is constant over the time-span of the data. We observed a distinct negative trend when plotting the total number of crimes in each year; hence, we also include time as a covariate. The required BAU level covariates may be constructed in an analogous fashion to the way in which the covariates were constructed in Section 4.2. Next, we generate spatio-temporal basis functions automatically using `auto_basis()`.

```
R> basis <- auto_basis(STplane(), chicago_crimes_fit, tunit = "years")
```

Then, we initialise and fit the '`SRE`' object using `FRK()`, setting `response = "poisson"` and `link = "log"`. Each entry of our data provides the location and time at which a given crime occurred. It also contains a column of ones called `"number_of_crimes"`; this will be used for binning. By default, `SRE()`, which is called internally within `FRK()`, bins and then averages data falling into the same BAU. We wish to model the total number of crimes in a given BAU; hence, we wish to sum the binned data instead of average. To do so, we pass the name of the

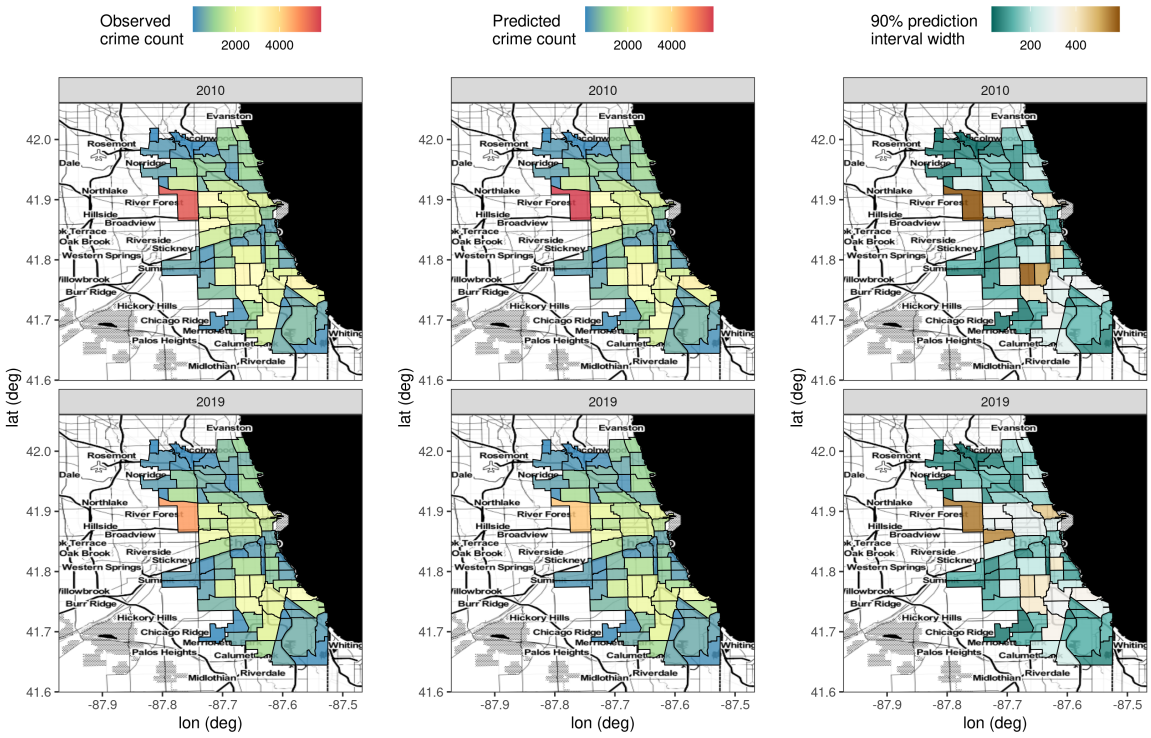

Figure 16: Observed number of crimes, predictions, and prediction uncertainty over Chicago in the prediction (2010) and forecast (2019) years. The first row corresponds to the year 2010; the second row corresponds to the year 2019. The first column shows the observed number of crimes; the second column shows the predicted number of crimes; the third column shows the width of a posterior predictive interval with a purported coverage of 90%.

response variable (`"number_of_crimes"`) via the argument `sum_variables`. As the number of spatial BAUs (the CAs) is relatively low, and we have observed each spatial BAU multiple times, we may attribute each spatial BAU its own fine-scale variance parameter (see Section 2.5).

```
R> S <- FRK(f = number_of_crimes ~ log(population) + x1 + x2 + x3,
+     data = list(chicago_crimes_fit), basis = basis, BAUs = ST_BAUs,
+     response = "poisson", link = "log",
+     sum_variables = "number_of_crimes", fs_by_spatial_BAU = TRUE)
```

Finally, we predict over the spatio-temporal BAUs (which use the CAs as spatial BAUs) using `predict()`, and plot the results using `plot()`.

To validate predictions, we excluded the years 2010 and 2019 from the training data. The observed number of crimes, predicted number of crimes, and prediction uncertainty for these years are provided in Figure 16. For both years, Figure 16 shows agreement between the predicted and observed number of crimes. Furthermore, the prediction uncertainty is roughly proportional to the predicted value, as one may expect when modelling counts. For these validation years, we also computed the empirical coverage when using 90%, 80%, 70%, and

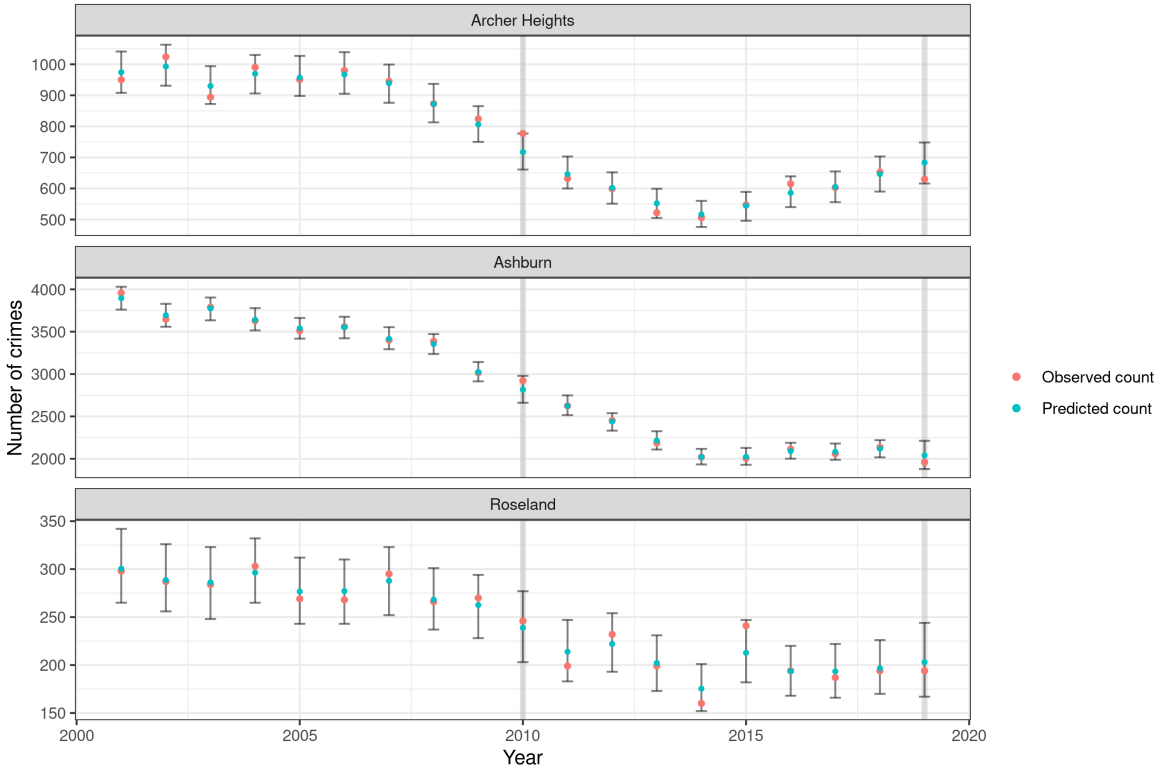

Figure 17: Time-series plots of predictions and observed number of crimes for three CAs of interest. The validation years, 2010 and 2019, are highlighted in light-grey. The observed number of crimes at each time is indicated by a red point, whilst the predicted number of crimes is indicated by a blue point. The error bars represent a 90% posterior predictive interval. We note that the posterior predictive intervals are slightly wider in validation years (2010 and 2019) than in observed years, and that the observed crime is contained within the predictive interval for all time-points for these CAs.

60% posterior predictive intervals, and the mean absolute percentage error (MAPE; see Appendix C). We consistently observed that the empirical coverage in the year 2010 was slightly (4% on average) higher than the purported coverage, whilst it was lower (16% on average) in the forecast year. We observed MAPE scores of 4.4% and 9.0% in the years 2010 and 2019, respectively. The slightly worse results in the year 2019 is possibly expected, as forecasting into the future is, in general, a harder task than predicting within the time-span of the data. Nonetheless, these predictions are cause for optimism given the difficulties in modelling crime in a spatio-temporal setting.

For illustration, we chose three CAs of interest: Ashburn, Roseland, and Archer Heights. The time-series of the observed data, predictions, and 90% posterior predictive intervals for these CAs is shown in Figure 17. The posterior predictive intervals are slightly wider in validation years (2010 and 2019) than in observed years. The observed number of crimes is contained within the predictive interval for all time-points for these CAs. The predictive distributions in the validation years for the three CAs of interest is shown in Figure 18. The forecasts for

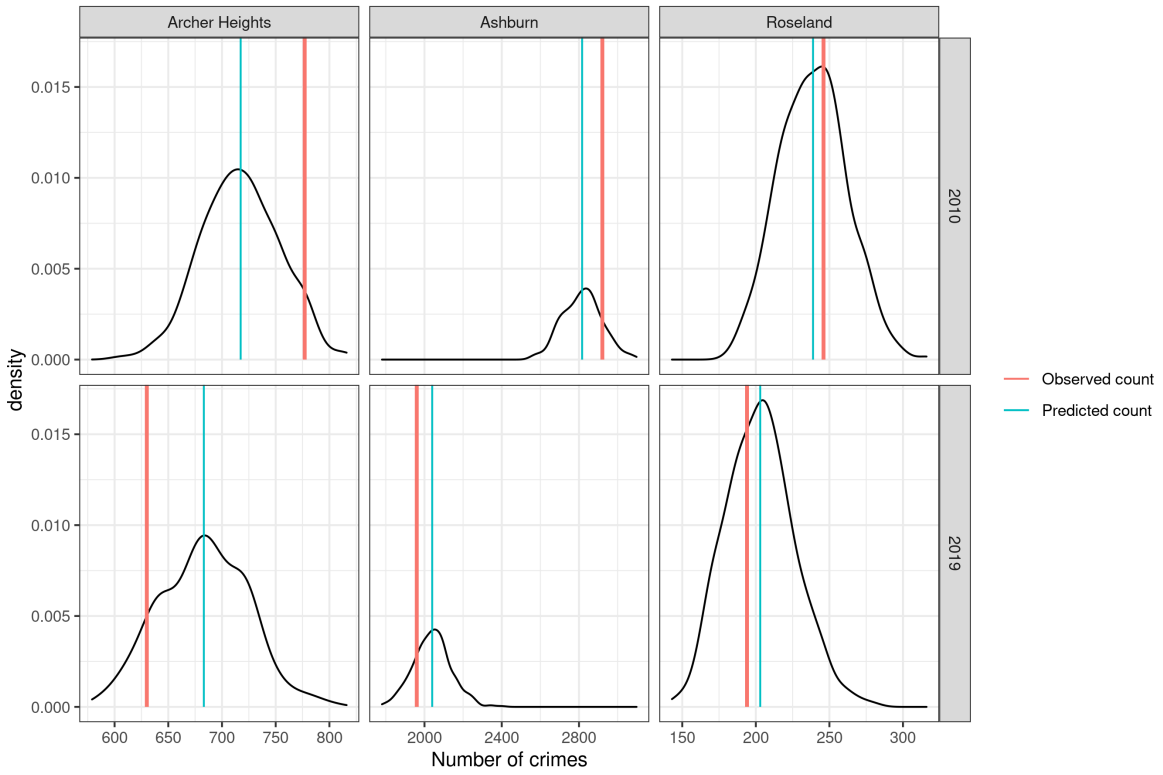

Figure 18: Posterior predictive distributions in the validation years (2010 and 2019) for three CAs (Archer Heights, Ashburn, and Roseland). The red and blue lines correspond to the observed and predicted number of crimes at each CA in the given year. The first *row* corresponds to the year 2010; the second row corresponds to the year 2019. The first *column* corresponds to Archer Heights; the second column corresponds to Ashburn; the third column corresponds to Roseland.

Ashburn and Roseland in the year 2019 are particularly accurate, with the predicted number of crimes essentially equal to the observed number of crimes.

5. Conclusion

In this paper we have described an extension to the R package **FRK** which allows for the spatial and spatio-temporal modelling and prediction of big, non-Gaussian data. Thanks to the use of a GLMM model and the software **TMB**, **FRK** v2 can now cater for many distributions within the exponential family, and many link functions. Furthermore, **FRK** v2 allows for the use of many more basis functions when modelling the spatial process, and can therefore also often achieve more accurate predictions in a Gaussian setting than **FRK** v1. The existing functionality of **FRK** is retained with this extension; in particular, the package makes use of automatic basis function construction, is capable of handling both point-referenced and areal data, and eases the so-called spatial change-of-support problem through the use of BAUs. The package 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.

One limitation of the framework is that it requires covariates to be known for every BAU, which may not be the case if covariates are recorded only at the data support level. Another limitation is the necessity to fix the fine-scale variance parameter in spatial change-of-support applications when `method = "TMB"`; note that this is not an issue if one is able to obtain a reliable estimate through other means (e.g., via previous census data). We are currently exploring avenues to address this limitation, either through adjusting the model to allow estimation of the fine-scale variance parameter within **TMB**, or via a more robust offline estimate. Another limitation is that despite the added flexibility, several models of interest, such as the zero-inflated Poisson, are still not catered for. The introduction of different types of models is facilitated by **TMB**'s implementation of automatic differentiation, which means we can make use of existing code within the C++ template straightforwardly; future work will see the introduction of other models of interest. The main spatial data structures used in **FRK** come from the package **sp** (Pebesma and Bivand 2005); future work may entail the inclusion of other spatial data structures, such as those from the package **sf** (Pebesma 2018).

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, and Michael Bertolacci for discussion surrounding the MODIS comparison study.

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. doi:10.1111/2041-210X.13168.

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: 2019-06-15.

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." https://www.cmap.illinois.gov/documents/10180/126764/_Combined_AllCCAs.pdf/. Accessed: 2020-09-18.

Cressie N (1993). *Statistics for Spatial Data*. revised edition. John Wiley & Sons, Hoboken, NJ.

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*, **70**, 209–226.

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*, **47**, 299–350.

Finley AO, Banerjee S, Gelfand AE (2015). "`spBayes` for large univariate and multivariate point-referenced spatio-temporal data models." *Journal of Statistical Software*, **63**, 1–28. URL http://www.jstatsoft.org/v63/i13/.

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. doi:10.1198/106186006X132178.

Gneiting T, Balabdaoui F, Raftery AE (2007). "Probabilistic forecasts, calibration and sharpness." *Journal of the Royal Statistical Society*, **69**, 243–268.

Guennebaud G, Jacob B, *et al.* (2010). "`Eigen` v3." http://eigen.tuxfamily.org. Accessed: 11-05-2019.

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.

Hu G, Bradley J (2018). "A Bayesian spatial-temporal model with latent multivariate log-gamma random effects with application to earthquake magnitudes." *Stat*, **7**, e179.

Huang C, Yao Y, Cressie N, Hsing T (2009). "Multivariate intrinsic random functions for cokriging." *International Association for 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**(2), 81–95. doi:10.32614/RJ-2014-026. URL https://doi.org/10.32614/RJ-2014-026.

Kassambara A (2020). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.4.0, URL https://CRAN.R-project.org/package=ggpubr.

Kristensen K, Nielsen A, Berg CW, Skaug H, Bell BM (2016). "TMB: Automatic differentiation and Laplace approximation." *Journal of Statistical Software*, **70**, 1–21.

Lee BS, Park J (2020). "A scalable partitioned approach to model massive nonstationary non-Gaussian spatial datasets." *arXiv:2001.09111*.

Lindgren F, Rue H (2015). "Bayesian spatial modelling with R-INLA." *Journal of Statistical Software*, **63**, 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: Series 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.

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 E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**, 439–446. doi:10.32614/RJ-2018-009. URL https://doi.org/10.32614/RJ-2018-009.

Pebesma EJ, Bivand RS (2005). "Classes and methods for spatial data in R." *R News*, **5**, 9–13. URL https://CRAN.R-project.org/doc/Rnews/.

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*, **71**, 3319–392.

Sengupta A, Cressie N (2013). "Hierarchical statistical modelling of big spatial datasets using the exponential family of distributions." *Spatial Statistics*, **4**, 14–44.

Sengupta A, Cressie N (2016). "Predictive inference for big, spatial, non-Gaussian data: MODIS cloud data and its change-of-support." *Australian & New Zealand Journal of Statistics*, **58**, 15–45.

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." https://www.lib.uchicago.edu/e/collections/maps/censusinfo.html. Accessed: 2020-09-15.

Urban Center for Computation and Data and University of Chicago (2020). "Plenario." http://plenar.io/explore/discover. Accessed: 11-09-2020.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wood S (2017). *Generalized Additive Models: An Introduction with R*. 2nd 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*, **In press**.

Zammit-Mangion A, Ng TLJ, Vu Q, Filippone M (2021). "Deep compositional spatial models." *Journal of the American Statistical Association*, **In press**.

Zhang B, Cressie N (2020). "Bayesian inference of spatio-temporal changes of Arctic sea ice." *Bayesian Analysis*, **15**(2), 605–631.

A. Parametrisations of the basis-function coefficients

Recall from Section 2.1 that **FRK** v2 allows the basis-function coefficients, $\boldsymbol{\eta}$, to be parametrised using either a prior covariance matrix, $\boldsymbol{K}$, or using a prior precision matrix, $\boldsymbol{Q}$. In this appendix, we describe these matrices. Recall that both formulations use block-diagonal matrices, wherein basis-function coefficients are independent of the basis-function coefficients in differing resolutions. Hence, we need only describe the intra-resolution dependencies.

A.1. Covariance matrix

Let $K_k(\boldsymbol{s}, \boldsymbol{s}^*)$ denote the covariance function of the basis-function coefficients corresponding to the kth basis function resolution. In **FRK**, we model $K_k(\boldsymbol{s}, \boldsymbol{s}^*)$ using the exponential covariance function,

$$K_k(\boldsymbol{s}, \boldsymbol{s}^*) = \sigma_k^2 \exp\left\{\frac{-d(\boldsymbol{s}, \boldsymbol{s}^*)}{\tau_k}\right\}, \tag{A.1}$$

where $d(\boldsymbol{s}, \boldsymbol{s}^*)$ is the distance between locations $\boldsymbol{s}, \boldsymbol{s}^* \in D$. Clearly (A.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 allows covariance tapering (Furrer, Nychka, and Genton 2006) of the intra-resolution covariance function. Noting that (A.1) is a special case of the Matérn covariance function with $\nu = 0.5$, we follow the recommendation of Furrer *et al.* (2006) and use the Spherical taper:

$$T_{\beta_k}(\boldsymbol{s}, \boldsymbol{s}^*) = \left\{1 - \frac{d(\boldsymbol{s}, \boldsymbol{s}^*)}{\beta_k}\right\}_+^2 \left\{1 + \frac{d(\boldsymbol{s}, \boldsymbol{s}^*)}{2\beta_k}\right\}, \tag{A.2}$$

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 define β_k based on the minimum distance between basis-function centroids; specifically, we set $\beta_k = \texttt{taper} \times \text{mindist}(k)$, where mindist$(k)$ is the minimum distance between the centroids of basis functions at the kth resolution and `taper` is a user-specified argument. The tapered covariance function is obtained by taking the product of the original covariance function (A.1) and the taper function (A.2).

A.2. Precision matrix

FRK v2 offers two types of sparse precision matrices: One for regularly spaced basis functions, and another for irregularly spaced basis functions. This choice is determined by the field `regular` in the '`Basis`' object.

When the basis functions are regularly spaced (`regular = TRUE`), **FRK** v2 uses a precision matrix that is related to that used in the R package **LatticeKrig** (Nychka, Hammerling, Sain, and Lenssen 2016). Let $\mathcal{N}_{i,k}$ denote the set of first-order horizontal and vertical neighbouring basis functions of the ith basis function of resolution k, and let $\boldsymbol{Q}_k$ denote the prior precision matrix of the basis-function coefficients at resolution k.

We model the elements of $\boldsymbol{Q}_k$ as

$$\{\boldsymbol{Q}_k\}_{i,j} = \begin{cases} \kappa_k + \rho_k |\mathcal{N}_{i,k}| & i = j \\ -\rho_k & j \in \mathcal{N}_{i,k} \\ 0 & \text{otherwise} \end{cases}, \tag{A.3}$$

where κ_k and ρ_k are parameters that need to be estimated. We note that $\boldsymbol{Q}_k$ 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. We also note that **LatticeKrig** uses $\boldsymbol{Q}_k^\top \boldsymbol{Q}_k$ as the precision matrix blocks.

To cater for irregularly spaced basis functions, **FRK** v2 also offers a sparse precision matrix which considers the distance between basis functions:

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

where κ_k, ρ_k, and κ_k are parameters that need to be estimated, and $T_{\beta_k}(\cdot, \cdot)$ is defined as in the preceding section. Again, this matrix is diagonally dominant, and hence positive definite. Equation (A.4) is in some ways a generalisation of (A.3). 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.

B. Distributions with size parameters

Two data models that can be used with **FRK** v2, namely, the binomial and negative-binomial distributions, have a known constant 'size' parameter k_j and a 'probability of success' parameter, π_j, associated with every datum Z_j. For binomial data models, k_j represents the number of trials, and Z_j the number of successes; for negative-binomial data models, k_j represents the target number of successes, and Z_j the number of failures.

Consider a negative-binomial data model with a logit-link function; under the standard interpretation of a link function (a function which transforms the mean $\mu(\cdot)$ to the linear predictor $Y(\cdot)$), one models

$$g(\mu(\cdot)) = \text{logit}(\mu(\cdot)) = Y(\cdot).$$

In this example, the range of the mean function (the inverse link function, $g^{-1}(\cdot)$) is $(0, 1)$. However, the mean of negative-binomial distribution may take values in $[0, \infty)$. Direct use of the logit link would thus unacceptably restrict the range of the mean function.

Therefore, for both the binomial and negative-binomial distributions, we first model $\pi(\cdot)$ as a function of $Y(\cdot)$, and then link $\pi(\cdot)$ to the mean $\mu(\cdot)$. That is, we use a hierarchical link function;

$$f(\pi(\cdot)) = Y(\cdot),$$
$$h(\mu(\cdot); k) = \pi(\cdot),$$

where $h(\cdot)$ is a function determined solely by the response distribution, and $f(\cdot)$ is a function that maps $\pi(\cdot)$ to the latent process $Y(\cdot)$. The implied link function is

$$g(\mu(\cdot); k) = f(h(\mu(\cdot); k)) = (f \circ h)(\mu(\cdot); k) = Y(\cdot).$$

Using a hierarchical link function approach with a negative-binomial data model and a logit-link function, we have that

$$f(\pi(\cdot)) = \text{logit}(\pi(\cdot)) = Y(\cdot),$$

and, as the expectation of the negative-binomial distribution in terms of the probability of success is $\mu(\cdot) = k\left(\frac{1}{\pi(\cdot)} - 1\right)$, we have that

$$h(\mu(\cdot)) = \frac{k}{\mu(\cdot) + k} = \pi(\cdot).$$

Observe that $\pi(\cdot) \in (0, 1)$, so that $\mu(\cdot) \in (0, \infty)$. Hence, in **FRK** v2, whenever the data model is specified to be binomial or negative-binomial, and a logit, probit, or complementary log-log 'link' is specified, we use it to define $f(\cdot)$ and to transform the probability parameter. We then map the probability parameter to the mean of the data using the known form of the mean specific to the distribution in question via $h(\cdot)$. If the user specifies a link function that is not appropriate for modelling probability parameters (such as the log or square-root link), then we use this to define $g(\cdot)$ directly, whilst also accounting for the size parameter; specifically, we set $g(\mu(\cdot)/k) = Y(\cdot)$.

C. Scoring rules

Suppose that we have a 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. For simplicity, we describe the measures in terms of prediction of the mean process.

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

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

- (Empirical) mean-absolute error (MAE): The mean-absolute error, also used to assess point-wise predictive performance, is

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

- (Empirical) mean-absolute percentage error (MAPE): The mean-absolute percentage error, is similar to the MAE, but we also divide by the true value;

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

- Continuous ranked probability score (CRPS; Gneiting *et al.* 2007, sec 4.2.): Let $F(\mu; \boldsymbol{s}, \boldsymbol{Z})$ denote the posterior predictive cumulative distribution function (CDF) of the mean process at location $\boldsymbol{s}$. The CRPS is used to evaluate a predictive CDF, and is defined as

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

  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-normal) 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).

- Interval score (Gneiting *et al.* 2007, sec. 6.2): The interval score for a purported $(1 - \alpha) \times 100\%$ prediction interval is defined as

$$S_\alpha^{\text{int}} \equiv \frac{1}{|D^*|} \sum_{\boldsymbol{s} \in D^*} \Bigg(u(\boldsymbol{s}) - l(\boldsymbol{s}) + \\ \frac{2}{\alpha}(l(\boldsymbol{s}) - \mu(\boldsymbol{s}))\mathbb{1}\{\mu(\boldsymbol{s}) < l(\boldsymbol{s})\} + \frac{2}{\alpha}(\mu(\boldsymbol{s}) - u(\boldsymbol{s}))\mathbb{1}\{\mu(\boldsymbol{s}) > u(\boldsymbol{s})\} \Bigg),$$

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

- Coverage: The coverage of a prediction interval is defined as

$$\text{Cvg} \equiv \frac{1}{|D^*|} \sum_{\boldsymbol{s} \in D^*} \mathbb{1}\{l(\boldsymbol{s}) \leq \mu(\boldsymbol{s}) \leq u(\boldsymbol{s})\}$$

  If the interval is indeed a $(1 - \alpha) \times 100\%$ prediction interval, the coverage should be approximately equal to $1 - \alpha$.

- Brier score (Gneiting *et al.* 2007, sec 3.): The Brier score, applicable in a binary setting, is defined as

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

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

D. Sydney poverty lines

Here we provide some details on how we defined the poverty line for the data in Section 4.3. Recall that our data consists of the number of families of various types (these types are 'couple family with no children', 'couple family with children','one parent family', and

'other family') within a range of weekly income brackets. For families with a weekly income below \$1000 (which, as we explain subsequently, are the families of interest for this analysis), the income brackets are defined in intervals of \$200: negative or nil income, [\$1–\$199], [\$200–\$399], ..., [\$800–\$999]. To determine the number of families 'in poverty' within each Statistical Area, we must define poverty lines. The Melbourne Institute of Applied Economic and Social Research (MIAESR) provided poverty line guidelines for a range of family structures in March 2011 (https://melbourneinstitute.unimelb.edu.au/assets/documents/poverty-lines/2017/Poverty-Lines-Australia-March-Quarter-2011.pdf). Unfortunately, the groupings of our family units do not align exactly with the poverty line definitions as given by the MIAESR, and so, since this example is shown for purely illustrative purposes, we make several assumptions. First, we assume 'families with children' consist of exactly two parents and two children. Second, since 'other families' 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, our 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 use the first half of Table 1 of the MIAESR guidelines. Fourth, our data do not provide exact income figures, but rather income brackets of width \$200; we thus round MIAESR guidelines to the nearest \$200. Hence, the definition of poverty lines (in Australian dollars) for each family unit considered in this study are the weekly incomes of: \$600 for a couple with no children, \$800 for a couple with children, and \$600 for a one parent family. We do not consider SA2 regions where the total number of families (the size parameter) is equal to zero, as this would cause issues with model fitting. (Note that there is nothing in the framework preventing us from *predicting* over these 'empty' SA2 regions, however, for interpretability reasons, we choose not to do so.)

Affiliation:

Matthew Sainsbury-Dale, Andrew Zammit-Mangion, Noel Cressie
National Institute for Applied Statistics Research Australia (NIASRA)
School of Mathematics and Applied Statistics
University of Wollongong
Wollongong, Australia
E-mail: msdale@uow.edu.au
URL: https://github.com/MattSainsbury-Dale