Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

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, and their analysis is needed in a variety of studies, such as those involving small-area demographics or global remote sensing instruments. **FRK** is an R package for spatial/spatio-temporal modelling and prediction with very large data sets that, to date, only supports linear process models and Gaussian data models. In this paper, we describe a major extension to **FRK** so that non-Gaussian data can be analysed in a generalised linear mixed model framework. The existing functionality of **FRK** is retained with this advance into non-linear, non-Gaussian models; in particular, it allows for automatic basis-function construction, it can handle both point-referenced and areal data simultaneously, and it can predict process values at any spatial support from these data. These vastly more general spatial and spatio-temporal models are fitted using the Laplace approximation via the software **TMB**. We demonstrate innovative features in this new version of **FRK**, highlight its ease of use, and compare it to alternative packages, using both simulated and real data sets.

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

1. Introduction

Non-Gaussian spatial and spatio-temporal data arise from a vast array of sources, including statistical studies in contaminated soil (e.g., Paul and Cressie 2011), global remote sensing (e.g., Sengupta and Cressie 2016), small-area demographics (e.g., Bradley, Wikle, and Holan 2016), and earthquake magnitudes (e.g., Hu and Bradley 2018). The statistical modelling of these 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/spatio-temporal 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 approximately unbiased 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 of distributions (McCullagh and Nelder 1989), and the mean is modelled using a transformation of some latent spatial process $Y(\cdot)$. Optimal prediction or estimation of unknown quantities from m observations entails the inversion of an $m \times m$ covariance matrix for many statistical models. Since this task is generally $O(m^3)$ in computational complexity, some form of dimension-reduction is required in 'big data' settings.

Reduced-rank variants of trans-Gaussian kriging are relatively under-developed (see Cressie, Sainsbury-Dale, and Zammit-Mangion (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 $Y(\cdot)$ a nearest neighbour Gaussian process (NNGP; Datta, Banerjee, Finley, and Gelfand 2016). 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 mutually independent local processes. 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 facilitate in a straightforward manner 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). Each of these packages have a different set of limitations: **spBayes**, **mgcv**, and **spNNGP** are limited to point-referenced data; **spBayes** uses basis functions that depend on covariance-function parameters, so computationally it can only handle a small number of predictive-process knots, resulting in 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 they cater for only a small number of non-Gaussian distributions. Further, none of these software packages cater for spatial change-of-support. Some general-purpose packages (e.g., **INLA**; Rue, Martino, and Chopin 2009; Lindgren and Rue 2015) can, in principle, handle

the wide array of modelling challenges posed by non-Gaussian spatial and spatio-temporal data; however, they are not specifically designed for this purpose and can be difficult for an unfamiliar user to implement. A project aimed at facilitating spatial statistical modelling using **INLA** is the **inlabru** package (Bachl, Lindgren, Borchers, and Illian 2019); at the time of writing, spatio-temporal modelling was not implemented in **inlabru**.

The package **FRK** (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 advance to **FRK** that allows it to cater for many distributions within the exponential family using the spatial GLMM framework; we henceforth refer to it as **FRK** v2 and the original version as **FRK** v1. **FRK** v2 provides a unifying framework that handles very large, spatial and spatio-temporal non-Gaussian data and, critically, 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 a few lines of intuitive, readable code. Optimal spatial prediction proceeds through the use of an *empirical* hierarchical statistical model (where likelihood-based estimates are substituted in place of unknown parameters) and a Monte Carlo (MC) algorithm, where a minimal number of user-level decisions is required. **FRK** v2 also accommodates the modelling of non-Gaussian spatial and spatio-temporal data on the surface of a sphere, a feature not offered by many other packages. Finally, although the primary motivation for this major upgrade is the modelling of non-Gaussian data, **FRK** v2 also allows for the use of substantially more basis functions than for **FRK** v1 when modelling the spatial process. Therefore, in a Gaussian setting, it often achieves more accurate predictions than **FRK** v1.

The remainder of the paper is organised as follows. In Section 2, we establish the statistical framework for **FRK** v2, and we describe model fitting and prediction using the R package **TMB** (Kristensen, Nielsen, Berg, Skaug, and Bell 2016). In Section 3, we discuss the new functionalities in **FRK** v2, provide a variety of 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 or spatio-temporal GLMM; specifically, 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 outline the approach of **FRK** v2 for spatio-temporal data.

2.1. The process layer

The process layer, which governs the conditional mean of the data, retains many similarities to that in **FRK** v1. Note that here we discuss the spatial case only; the extension to a

spatio-temporal setting is outlined in Section 2.5.

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 type of spatial variability. First, spatially referenced covariates $\boldsymbol{t}(\cdot)$ and their associated regression parameters $\boldsymbol{\alpha}$, capture spatial variation that is linked to known, usually large-scale, explanatory variables that are elements of $\boldsymbol{t}(\cdot)$; 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 the scales of spatial variation with just $\boldsymbol{t}(\boldsymbol{s})^\top \boldsymbol{\alpha}$ and $v(\boldsymbol{s})$ can result in overly optimistic predictions; this problem is alleviated by also including a fine-scale-variation random process, $\xi(\cdot)$.

In **FRK** v1/v2, the medium-to-small-scale term $v(\cdot)$ is constructed as a linear combination of r spatial basis functions with random coefficients, where r is fixed and smaller than m, the number of observations. 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 $\boldsymbol{\phi}(\cdot) \equiv (\phi_1(\cdot), \ldots, \phi_r(\cdot))^\top$ of pre-specified spatial basis functions. See Zammit-Mangion and Cressie (2021) for details on how these basis functions are constructed. The fine-scale term, $\xi(\cdot) \equiv \{\xi(\boldsymbol{s}) : \boldsymbol{s} \in D\}$, is modelled as white noise after discretisation, which we discuss next.

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 easily consider point-referenced and areal data simultaneously, and they facilitate solutions to spatial change-of-support problems. Then let $Y(A_i)$ denote a representative value of $\{Y(\boldsymbol{s}) : \boldsymbol{s} \in A_i\}$, where commonly the spatial integral or the spatial average over A_i is chosen. Define the discretised latent spatial process $Y(\cdot)$ evaluated over the N BAUs as $\boldsymbol{Y} \equiv (Y_1, \ldots, Y_N)^\top$, where $Y_i \equiv Y(A_i)$, $i = 1, \ldots, N$. Then, a vectorised version of (1) is

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

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

As in **FRK** v1, the elements of $\boldsymbol{\xi}$ are modelled as independent and identically distributed (i.i.d.) Gaussian random variables with variance σ^2_ξ, and $\boldsymbol{\eta}$ is modelled as a mean-zero multivariate-Gaussian random variable with covariance matrix $\mathrm{cov}(\boldsymbol{\eta}, \boldsymbol{\eta})$. 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 basis-function resolutions are independent; see Appendix A for how the intra-resolution dependencies with $\boldsymbol{K}$ and $\boldsymbol{Q}$ are modelled. In a non-Gaussian setting (when **TMB** is used for model fitting), use of $\boldsymbol{Q}$ instead of $\boldsymbol{K}$ is helpful 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 a mean process, $\mu(\cdot)$ (that we consider in more detail in Section 2.2):

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

Therefore, the mean process evaluated over the BAUs is $\boldsymbol{\mu} \equiv (\mu_i : i = 1, \ldots, N)^\top$, where $\mu_i = g^{-1}(Y_i)$, $i = 1, \ldots, N$, and $g^{-1}(\cdot)$ is the inverse link function.

Finally, we note that two distributions considered in this framework, namely, the binomial distribution and the negative-binomial distribution, have a known constant 'size' parameter and a 'probability of success' parameter associated with every datum. For these distributions, we also define a probability process, $\pi(\cdot)$, and its discretised version evaluated over the BAUs, $\boldsymbol{\pi}$; see Appendix B for further details.

2.2. The data layer

We denote the vector of m observations (the data vector) as $\boldsymbol{Z} \equiv (Z_1, \ldots, Z_m)^\top$, and each datum is originally associated with a spatial support, R_j, $j = 1, \ldots, m$, which we associate to one or more BAUs. In practice, these spatial supports may not coincide with entire BAUs and, when this is the case, we assume that a spatial support contains a BAU if and only if there is a non-empty intersection between the BAU and the spatial support[1]: That is, the indices of the BAUs associated with R_j are $c_j \equiv \{i : A_i \cap R_j \neq \varnothing\}$, for $j = 1, \ldots, m$. We then define the set of observation supports as $D^O \equiv \{B_j : j = 1, \ldots, m\}$, where $B_j \equiv \cup_{i \in c_j} A_i$. Since we approximate R_j with B_j, $j = 1, \ldots, m$, the BAUs need to be sufficiently fine when the data is point-referenced. Figure 1 shows a pedagogical example illustrating the relationship between the continuous domain D, the BAUs, the original spatial supports, and the observation supports.

Define the mean of the observations 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

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

where $\mathbb{I}(\cdot)$ is the indicator function, such that[2]

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

In **FRK** v2, the weights w_{ij} 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 to a *relative* weight v_i, $i = 1, \ldots, N$, such that $w_{ij} \propto v_i$, where the constant of proportionality can vary with j. For example, if the BAUs are of unequal area, then one may wish to set $v_i = |A_i|$. By default (and implicit in **FRK** v1), each v_i is set to 1. The argument `normalise_wts` controls whether $\boldsymbol{C}_Z$ corresponds to a weighted sum or a weighted average. If set to `FALSE`, then $w_{ij} = v_i$ for all j (weighted sum); if set to `TRUE` (implicit in **FRK** v1),

[1] **FRK** v1 assumed that a spatial support contains a BAU if and only if the BAU centroid lies within the spatial support; this modification allows **FRK** v2 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.

[2] Note that in **FRK** v1, Zammit-Mangion and Cressie (2021) applied $\boldsymbol{C}_Z$ directly to $\boldsymbol{Y}$: With the identity link function, implicit in **FRK** v1, $\boldsymbol{\mu}$ and $\boldsymbol{Y}$ are equivalent.

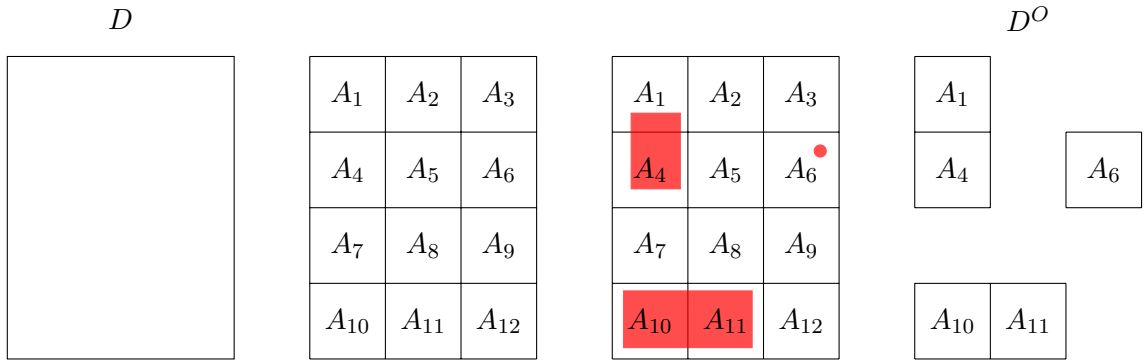

Figure 1: An illustration of how the continuous spatial domain, D, is discretised into the set of BAUs, and how the observation domain, D^O, is derived from the observation supports. (Left panel) The continuous spatial domain, D. (Centre-left panel) The spatial domain discretised into $N = 12$ BAUs. (Centre-right panel) The discretised domain superimposed with $m = 3$ observations, two of which are areally-referenced (C_1 and C_3), and one that is point-referenced (C_2). (Right panel) The observation domain, D^O: The observation supports that comprise D^O are $B_1 \equiv A_1 \cup A_4$, $B_2 \equiv A_6$, and $B_3 \equiv A_{10} \cup A_{11}$. See if I can add annotations for the B_j. Also possibly reduce the size of the letter A_i. Add R_j.

then the w_{ij} are normalised so that each row of $\boldsymbol{C}_Z$ sums to 1 (weighted average). Note that if $v_i = |A_i|$ and `normalise_wts = TRUE`, the normalised weights are $w_{ij} = |A_i|/|B_j|$, since $\sum_{i \in c_j} |A_i| = |B_j|$.

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

$$[Z_j \mid \mu(\cdot), \psi] = \text{EF}(\mu_{Zj}, \psi), \tag{5}$$

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, with the parameterisations assumed by **FRK** v2, $\psi = 1$ 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 means elsewhere in the domain. Further, we assume that all observations are conditionally independent given the latent spatial process: Specifically,

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

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

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

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

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 used.

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

$$Z_j \mid \mu_{Zj}, \psi \overset{\text{ind}}{\sim} \text{EF}(\mu_{Zj}, \psi), \quad j = 1, \ldots, m, \tag{7}$$

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

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

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

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

$$\boldsymbol{\xi} \mid \sigma^2_\xi \sim \text{Gau}(\boldsymbol{0}, \sigma^2_\xi \boldsymbol{V}), \tag{12}$$

where $\boldsymbol{V}$ is a known, positive-definite diagonal matrix, and σ^2_ξ is either unknown and estimated or it is possible for the user to provide it. In a spatio-temporal setting, a more complex model for $\boldsymbol{\xi}$ is allowed; see Section 2.5. Note that **FRK** v2 is backwards compatible, since an identity link function and a Gaussian data model in (7), where ψ plays the role of the measurement-error variance, yields the model used in **FRK** v1. When the data model is binomial or negative-binomial and a 'link' function is chosen that is appropriate for modelling probabilities, (9) is replaced with

$$h(\boldsymbol{\mu}; \boldsymbol{k}) = \boldsymbol{\pi}, \tag{13}$$

$$f(\boldsymbol{\pi}) = \boldsymbol{Y}, \tag{14}$$

where $h(\cdot;\cdot)$ and $f(\cdot)$ are discussed in greater detail in Appendix B.

2.3. Estimation

We now derive the likelihood functions required for model fitting, outline the intractable integrals that arise when non-Gaussian data models are fitted, 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

$$L(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}) \equiv [\boldsymbol{Z}, \boldsymbol{\eta}, \boldsymbol{\xi} \mid \boldsymbol{\theta}] = [\boldsymbol{Z} \mid \boldsymbol{\mu}_Z, \psi][\boldsymbol{\eta} \mid \boldsymbol{\vartheta}][\boldsymbol{\xi} \mid \sigma^2_\xi], \tag{15}$$

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 (15). 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 observed-data 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 (15):

$$L^*(\boldsymbol{\theta}; \boldsymbol{Z}) \equiv \int_{\mathbb{R}^p} L(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u}) \mathrm{d}\boldsymbol{u}, \tag{16}$$

where p is the total number of random effects in the model. When the data are non-Gaussian, the integral in (16) is typically intractable and must be approximated. In **FRK** v2, we use a Laplace approximation, which we now briefly 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 (15) that has the form of an un-normalised Gaussian density in terms of $\boldsymbol{u}$, with mean vector $\hat{\boldsymbol{u}}$ and covariance matrix $\boldsymbol{H}$. Substitution of this approximation into (16), and evaluation of the integral, yields the Laplace approximation of the observed-data 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 $[\boldsymbol{u} \mid \boldsymbol{Z}, \boldsymbol{\theta}] \propto [\boldsymbol{u}, \boldsymbol{Z} \mid \boldsymbol{\theta}]$, which 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 an un-normalised Gaussian density in terms of $\boldsymbol{u}$, it follows that, approximately, $\boldsymbol{u} \mid \boldsymbol{Z}, \boldsymbol{\theta} \sim \text{Gau}(\hat{\boldsymbol{u}}, \boldsymbol{H})$. In the software we use (**TMB**; see below), estimates of $\hat{\boldsymbol{u}}$ and $\boldsymbol{H}^{-1}$ are provided, which makes prediction of $\boldsymbol{u}$ and any function of it straightforward via MC simulation (see Section 2.4).

*Model fitting with **TMB***

FRK v2 supplies the R package **TMB** (Kristensen *et al.* 2016) with a C++ template function that defines $l(\boldsymbol{\theta}; \boldsymbol{Z}, \boldsymbol{u})$. **TMB** then computes the Laplace approximation of the observed-data log-likelihood, and it automatically computes its derivatives; these quantities are then invoked via a user-defined optimising function (`nlminb()` is used by default). **TMB** uses **CppAD** (Bell 2005) for automatic differentiation, and it uses the linear algebra libraries **Eigen** (Guennebaud, Jacob *et al.* 2010) and **Matrix** (Bates, Maechler, and Davis 2019) for vector and matrix operations in C++ and R, respectively. Use of these packages yields high computational efficiency. **TMB**'s implementation of automatic differentiation is a key reason why **FRK** v2 can easily cater for a large variety of response distributions and link functions, as each response-distribution/link-function combination does not need to be considered on a case-by-case basis.

Note that all unknown quantities are treated as random in **TMB** (with a flat prior assumed if a prior is not provided). To retain **FRK** v1's mixed-model interpretation, we fix the parameters and fixed effects to their posterior-mode estimates, and then we 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 quantities that could be of interest to the user, namely the latent process $Y(\cdot)$, the mean process $\mu(\cdot)$, and data at unobserved locations. 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 $\boldsymbol{u}$, approximate inference on $Y(\cdot)$ can be done using well-known formulas. However, the posterior distribution of a non-linear function of $Y(\cdot)$ (e.g., the mean $\boldsymbol{\mu}$) is typically not available in closed form, and some approximation is required. In **FRK** v2, we use a Monte Carlo (MC) approach to inference of $\boldsymbol{u} = (\boldsymbol{\eta}^\top, \boldsymbol{\xi}^\top)^\top$.

Recall 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

$$\boldsymbol{Y}_{\text{MC}} \equiv \boldsymbol{T}\boldsymbol{A} + [\boldsymbol{S}\ \boldsymbol{I}]\,\boldsymbol{U}, \tag{17}$$

where every column of the matrix $\boldsymbol{A}$ is 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 the N-dimensional vector $\boldsymbol{\mu}$ from $[\boldsymbol{\mu} \mid \boldsymbol{Z}, \boldsymbol{\theta}]$ via $\boldsymbol{M} \equiv g^{-1}(\boldsymbol{Y}_{\text{MC}})$, where $g^{-1}(\cdot)$ is applied element-wise. Finally, MC samples of data over all N BAUs, $\boldsymbol{Z}^* \equiv (Z_1^*, \ldots, Z_N^*)$, where $Z_i^* \sim \text{EF}(\mu_i, \psi)$, $i = 1, \ldots, N$, can be constructed straightforwardly using $\boldsymbol{M}$.

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

Arbitrary prediction regions

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

Prediction over D^P requires some form of aggregation across the associated BAUs. Since aggregation must be done on the response scale, we restrict prediction over arbitrary regions to the mean process and the data process. Let $\boldsymbol{\mu}_P \equiv \{\mu(\tilde{B}_k) : k = 1, \ldots, N_P\}$ be the mean process evaluated over the prediction regions. Just as $\boldsymbol{\mu}_Z$ was constructed from the BAU-level mean process $\boldsymbol{\mu}$ via the matrix $\boldsymbol{C}_Z$ given by (3), since each $\tilde{B}_k$ is a BAU or a union of BAUs, one can construct an $N_P \times N$ matrix

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

such that

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

As in Section 2.2, the proportionality constants, $\{v_i : i = 1, \ldots, N\}$, for the weights, $\{\tilde{w}_{ik}\}$, 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, $\{v_i : i = 1, \ldots, N\}$, and the same setting of `normalise_wts` are used in construction of both $\boldsymbol{C}_Z$ and $\boldsymbol{C}_P$.[3]

[3] Noel: "If you use `normalise_wts = FALSE` and sum, you will 'double count' when regions overlap". I don't think this is a problem, because the BAUs (the units over which the aggregation occurs) are defined to be non-overlapping. If prediction regions do overlap, we *want* to double count (think of the extreme case in which two prediction regions are exactly the same; they should return the same number).

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 from $[\boldsymbol{\mu} \mid \boldsymbol{Z}, \boldsymbol{\theta}]$. Predictions and uncertainty quantification of the predictions can then be computed straightforwardly from $\boldsymbol{M}_P$. When one wishes to predict data over aggregations of BAUs, that is, $\boldsymbol{Z}_P^* \equiv (Z_{P1}^*, \ldots, Z_{PN_P}^*)$, where $Z_{Pk}^* \sim \text{EF}(\mu(\tilde{B}_k), \psi)$, $k = 1, \ldots, N_P$, **FRK** v2 simulates from the specified response distribution with mean equal to the samples in $\boldsymbol{M}_P$, and then it returns predictions and uncertainty quantification of these simulated data.

2.5. Spatio-temporal framework

As in **FRK** v1, we accommodate spatio-temporal data by using spatio-temporal basis functions constructed via a tensor product of spatial and temporal basis functions. Since often one requires several thousand basis functions in a spatio-temporal setting, here we focus on the case where one models $\boldsymbol{\eta}$ using the sparse precision matrix $\boldsymbol{Q}$.

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 the 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 the $r_t r_s$ basis functions as

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

where $\otimes$ is the Kronecker product. This form for $\boldsymbol{Q}$ leads to significant computational savings. For the random coefficients associated with the temporal basis functions, **FRK** v2 uses an AR1 model.

In a spatio-temporal setting, it is possible that each spatial BAU is observed multiple times. It is also possible that the variance of the fine-scale component is not constant over the spatial domain D. In these situations where the temporal aspect gives repeat spatial observations, a better fit to the data 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 $\boldsymbol{\xi} \sim \text{Gau}(\boldsymbol{0}, \sigma_\xi^2 \boldsymbol{I})$, **FRK** v2 also allows the model, $\boldsymbol{\xi} \sim \text{Gau}(\boldsymbol{0}, \boldsymbol{\Sigma}_\xi)$, where

$$\boldsymbol{\Sigma}_\xi \equiv \begin{pmatrix} \text{diag}(\boldsymbol{\sigma}_\xi^2) & 0 & \ldots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & & \\ 0 & \ldots & & \text{diag}(\boldsymbol{\sigma}_\xi^2) \end{pmatrix} \tag{18}$$

is an $N \times N$ matrix, $\boldsymbol{\sigma}_\xi^2 \equiv (\sigma_{\xi,1}^2, \ldots, \sigma_{\xi,N_s}^2)^\top$, 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()` function. It is particularly useful when the number of spatial BAUs (and hence the number of variance parameters to estimate) is relatively low, and when we have observations from each spatial BAU at many time-points; see, for instance, the example presented in Section 4.4.

3. New features and their usage

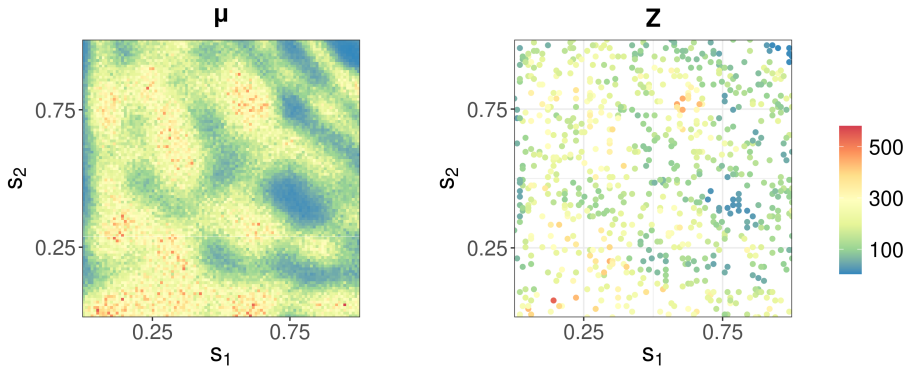

Figure 2: Simulated, point-referenced, Poisson data set used in the illustrative example of Section 3.1. (Left panel) True mean process, $\mu(\cdot)$, evaluated over the BAUs. (Right panel) Simulated Poisson data set.

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 package's 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 with non-Gaussian spatial point-referenced and area-referenced data, respectively. In Section 3.3, we show the potential improvement in predictive performance of **FRK** v2 over **FRK** v1 when the data are Gaussian, thanks to an increased maximum number of basis functions in **FRK** v2. Finally, in Section 3.4, we briefly discuss extensions available in **FRK** v2 to model data in a spatio-temporal setting. All results presented in the remainder of this paper can be generated using the reproducible code 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 at spatial locations shown in Figure 2. The true mean process evaluated over the BAUs, $\boldsymbol{\mu}$, is also 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 constructed 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()`. (Somewhere in this paragraph, we need to write down a formula for the Poisson distribution and the link function, etc.) For $j = 1, \ldots, m$, $Z_j \sim \text{Pois}(\mu_{Z_j}, \psi)$ and $g(\cdot) = \log(\cdot)$. These steps may also be performed in a single line of code with the convenient wrapper function `FRK()`. Note that when the data are non-Gaussian or when a non-identity link function is chosen, `FRK()` automatically enforces `method = "TMB"` and selects `K_type = "precision"`, which means that the basis-function coefficients are modelled via a sparse precision matrix $\boldsymbol{Q}$.

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

Prediction is done using `predict()`. The argument `type` specifies the quantities of interest

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

Function	Argument	Use
`SRE()`	`response`	A string indicating the response distribution.
	`link`	A string indicating the link function.
	`K_type`	A string indicating the parameterisation of cov($\boldsymbol{\eta}$, $\boldsymbol{\eta}$); the newly permissible value, `"precision"`, indicates that a sparse precision matrix should be used.
	`normalise_wts`	A flag controlling whether the weights in $\boldsymbol{C}_Z$ and $\boldsymbol{C}_P$ should be normalised or not.
	`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: `"TMB"` is required whenever a non-Gaussian data model or non-identity link function is used.
	`known_sigma2fs`	A positive number at which to fix the fine-scale variance.
`predict()`	`type`	A vector of strings indicating the quantities of interest for which inference is made. The inclusion of `"link"` indicates that inference on the latent process ($\boldsymbol{Y}$) is made; the inclusion of `"mean"` indicates that inference on the mean process ($\boldsymbol{\mu}$ or $\boldsymbol{\mu}_P$) and, if applicable, the probability process ($\boldsymbol{\pi}$) is made; and the inclusion of `"response"` indicates that inference on the data ($\boldsymbol{Z}^*$ or $\boldsymbol{Z}^*_P$) is made.
	`percentiles`	Numeric vector indicating the percentiles of the predictive distribution(s) to be returned.
	`n_MC`	Integer indicating the number of MC samples at each BAU.
`auto_BAUs()`	`spatial_BAUs`	The spatial BAUs in a spatio-temporal setting, where spatio-temporal BAUs are constructed by taking the Kronecker product of the spatial BAUs and temporal BAUs (box functions). If `NULL`, the spatial BAUs are constructed automatically from the data.
`plot()`	-	A method for visualising the data, predictions, and uncertainty quantification of the predictions given an '`SRE`' object and the '`Spatial*DataFrame`' or '`STFDF`' object resulting from 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 values that the data may take, 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					✓

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 predictive distributions and hence of prediction intervals; by default, the 5th and 95th percentiles are computed:

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

When `method = "TMB"`, the returned object from `predict()` is a '`list`' containing two elements. The first element is an object of the same class as `newdata` (if `newdata` is unspecified, prediction is done over the BAUs) and contains the predictions and uncertainty quantification of the predictions for each term in `type`. The second element is a '`list`' of matrices containing MC samples for each term in `type` at each prediction location. Finally, a '`list`' of '`ggplot`' (Wickham 2016) objects of the predictions and their associated uncertainty can be generated using the function `plot()`:

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

The '`ggplot`' objects can be arranged easily on a grid using various dedicated packages; we used **ggpubr** (Kassambara 2020). Figure 3 shows predictions and prediction-interval widths for the latent process $Y(\cdot)$ and the mean process $\mu(\cdot)$ ($\boldsymbol{Y}$ and $\boldsymbol{\mu}$). The predictions of $\mu(\cdot)$ are reasonable given the data and the true process shown in Figure 2. The prediction-interval widths for $Y(\cdot)$ overall, do not vary much, but they are larger in regions of data paucity and along the boundary of the spatial domain; on the other hand, the prediction-interval width for $\mu(\cdot)$ is large when $\mu(\cdot)$ is large, as can be expected when the response is Poisson distributed. The 'bullseye' points of low uncertainty, visible for both processes, correspond to the data locations. The point-like nature of this reduction in uncertainty arises from the

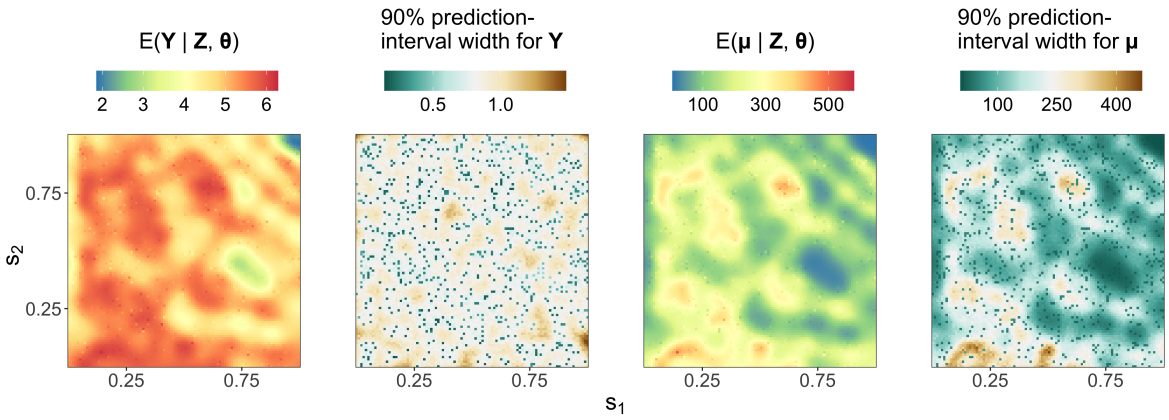

Figure 3: Predictions and prediction-interval widths using the data shown in Figure 2. (Left panel) Prediction of the latent process, $Y(\cdot)$. (Centre-left panel) Width of the 90% central prediction interval for $Y(\cdot)$. (Centre-right panel) Prediction of the mean process, $\mu(\cdot)$. (Right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

fine-scale random effects, $\boldsymbol{\xi}$ (or $\xi(\cdot)$??), 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 areas with large spatial supports. Again, I need to give a formula for the negative binomial distribution, link function, etc.. I think Noel is saying that, in all of these examples, it would be helpful to write: "$Z_j \sim \text{NB}(\boldsymbol{\mu}_{Z,j}, \psi)$, $g(\cdot) = \text{logit}(\cdot)$, etc."

In the the first step of data simulation, 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 both areal-level and BAU-level observations. We construct the latent 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; 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

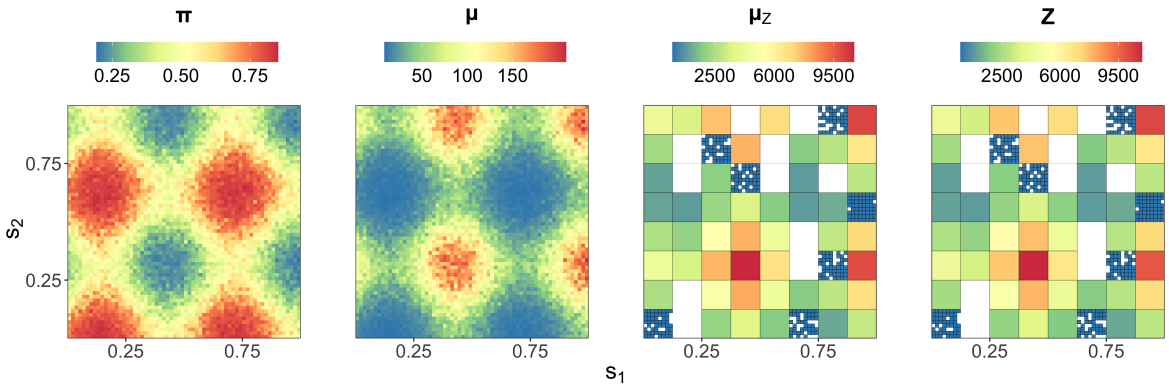

Figure 4: The simulated, areal, negative-binomial data set used in the illustrative example of Section 3.2. (Left panel) True probability process evaluated over the BAUs. (Centre-left panel) True mean process evaluated over the BAUs. (Centre-right panel) True mean process aggregated to the data-support level. (Right panel) Simulated data at the data-support level, with some observations omitted; these are the data used for model fitting. Need a better explanation of this figure. Noel is confused about missing data at the BAU level. Add something like: "In this example, we have both large-scale and fine-scale observations: That is, some of the data are measured over large areal regions comprising many BAUs, while other data have supports equal to the BAUs." Might also change the axis labels like I did in Figure 3.

need the size parameter of every observed BAU[4]. 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 either through the observations or through the BAUs. When some observations are associated with multiple BAUs, the user must provide the size parameter at the BAU level (for all observed BAUs). (So how are individual BAU size parameters used to get a size parameter for areal-level supports? I think I need to tell the reader that, in the case of binomial and negative-binomial data, we restrict the $\boldsymbol{C}_Z$ to correspond to a "simple sum"; that is, all elements non-zero elements of $\boldsymbol{C}_Z$ are enforced to be 1. Then, the elements of $\boldsymbol{\mu}_Z$ are obtained as a simple sum over the associated elements of $\boldsymbol{\mu}$, and it is reasonable to construct the size parameter for areal-level supports as the sum over BAU size parameters.)

```
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 $\pi(\cdot)$ and $\mu(\cdot)$ over the BAUs.

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

[4]Noel crossed out "observed BAU" and replaced it with "observation". I disagree: Since our model requires the construction of $\boldsymbol{\mu}$ (actually, a subset of $\boldsymbol{\mu}$ over only those BAUs associated with an observation), we need the size parameter of every BAU associated with an observation.

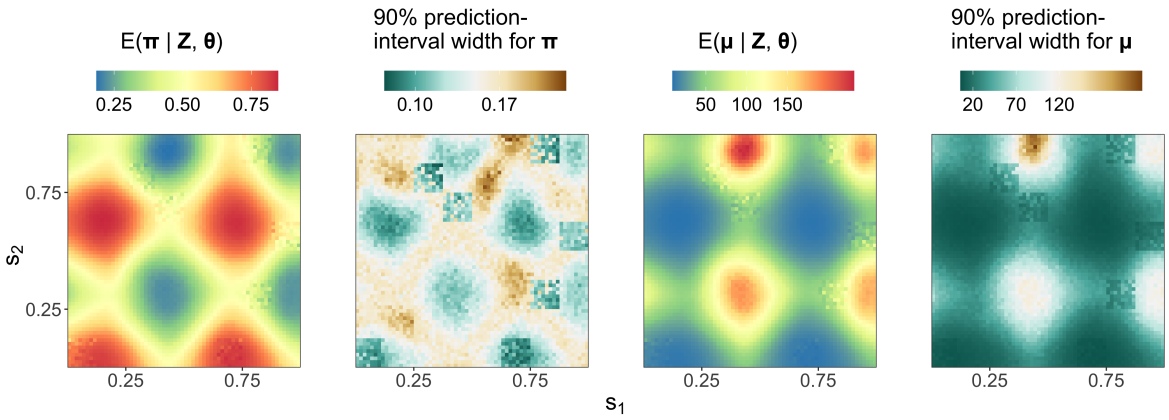

Figure 5: Prediction and prediction-interval width for the simulated negative-binomial areal data set shown in the right panel of Figure 4. (Left panel) Prediction of the probability process, $\pi(\cdot)$. (Centre-left panel) Width of the 90% central prediction interval for $\pi(\cdot)$. (Centre-right panel) Prediction of the mean process, $\mu(\cdot)$. (Right panel) Width of the 90% central prediction interval for $\mu(\cdot)$.

Figure 5 shows the predictions and prediction-interval widths for both the mean process, $\mu(\cdot)$, and the probability process, $\pi(\cdot)$, at the BAU level. ($\pi(\cdot)$ is not explained, need to give a formula that includes $\pi(\cdot)$ at the start of this section.) We observe agreement between the fields shown in Figure 4 and the corresponding predictions. The prediction-interval width for $\mu(\cdot)$ is roughly proportional to its prediction. In contrast, the prediction-interval width for $\pi(\cdot)$ is low when the prediction is near 0 or 1, and it 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% prediction intervals was 90.9%, which is almost nominal and very reasonable given the inherent difficulty with spatial change-of-support.

To emphasise that the prediction polygons are unrelated to the BAUs and data supports (but don't we use BAUs to work with discretised polygons? Perhaps I should say: "... that the shapes of the prediction polygons are unrelated to those of 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 do not allow predictions of $\pi(\cdot)$ over arbitrary polygons. (Where do I say this in Section 2.4? I either need to add a sentence in Section 2.4 explicitly saying that we cannot predict $Y(\cdot)$ and $\pi(\cdot)$ over arbitrary prediction regions, or I need to change this section slightly.) Figure 6 shows the predictions and prediction-interval widths for $\mu(\cdot)$ over the irregularly-shaped areas.

3.3. Increased numbers of basis functions in v2

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

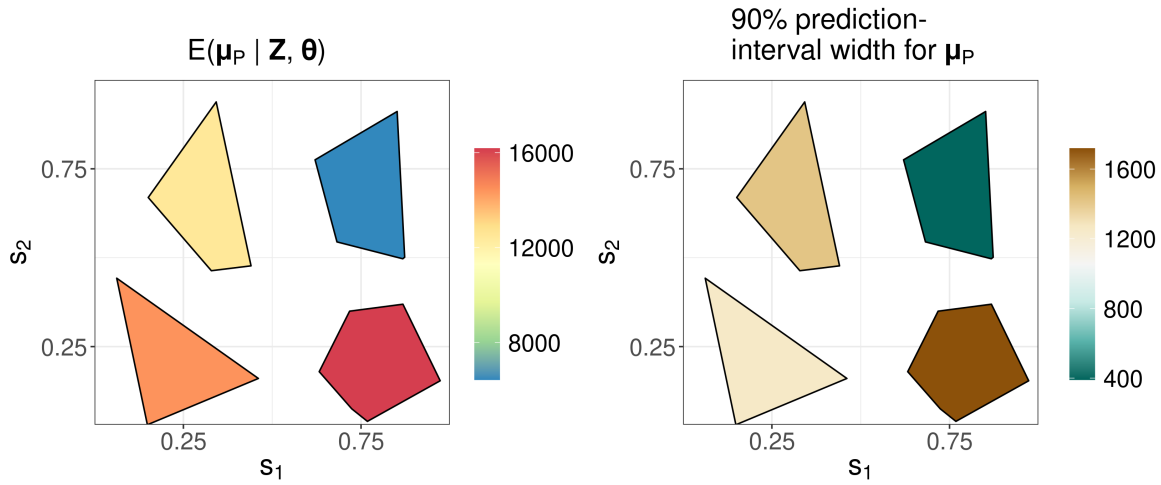

Figure 6: Prediction (left panel) and prediction-interval widths (right) for the aggregated mean, $\boldsymbol{\mu}_P$, when predicting over arbitrary polygons in the example of Section 3.2. Note that these polygons have equal area.

Table 3: Diagnostics comparing the predictive performance when using a range of basis-function resolutions with point-referenced count data. The diagnostics are the root-mean-squared prediction error (RMPSE), the continuous ranked probability score (CRPS), and the empirical coverage (Cvg90) and interval score (IS90) resulting from a central prediction interval with a nominal coverage of 90% (see Appendix C for further details). The diagnostics are with regard to prediction of the BAU-level mean, $\boldsymbol{\mu}$, at all unobserved locations and they are averaged.

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

performance of the framework can often be determined by the number of basis functions, as shown in the following examples.

First, 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 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 presence of the fine-scale random variation term, $\xi(\cdot)$, in (1).

Second, we re-ran the analysis for the comparative study published in Heaton *et al.* (2019). The data used in that study was made up of a training data set and a test data set consisting of 105,569 observations and 42,740 observations, respectively. 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 used 12,114. The results show that the increased number of basis

Table 4: Scores 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	1.38	1.81	0.98	9.02	0.89	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 its run time is not directly comparable. **FRK** v2 was implemented using a machine with 16 GB of RAM and an Intel i7-9700 3.00GHz CPU with 8 cores. The other models were implemented using the 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.

functions significantly improves the diagnostic scores over **FRK** v1, so that the results for **FRK** v2 are now comparable to the heretofore-preferred 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.

3.4. Spatio-temporal extensions

FRK v2 also caters for non-Gaussian, point-referenced and areally-referenced, *spatio-temporal* data. (Noel suggests "point-referenced and areally-referenced" should be "BAU-level and areal-level"; I disagree, because the data itself is originally point-referenced, irrespective of the fact that we associate it to BAUs.) For the sake of brevity, we will not use a simple example to show this, and instead we refer readers to the application given in Section 4.4.

4. Application and comparison studies

We now provide several application and comparison studies using **FRK** v2. In Section 4.1, we present a comparison study between **FRK** v2 and other packages that cater for non-Gaussian data models. In Section 4.2, we demonstrate block prediction using contaminated soil data and compare our results to that from another package. In Section 4.3, we use

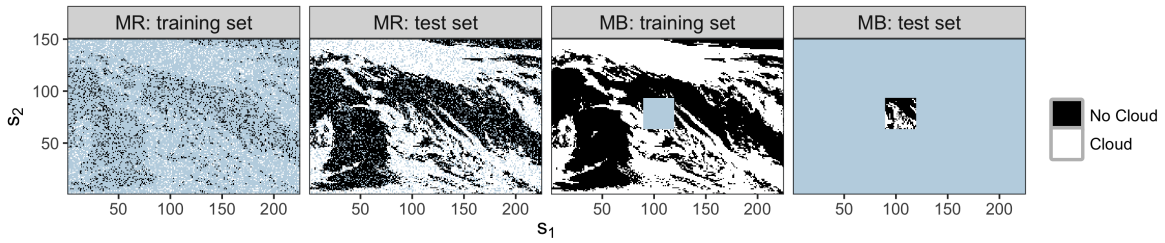

Figure 7: MODIS data used in the comparative study of Section 4.1; a blue background is used to denote data removed. (Left panel) The missing-at-random (MR) sample used for training. (Centre-left panel) The MR test data. (Centre-right panel) The 'missing-in-a-block' (MB) sample used for training. (Right panel) The MB test data.

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

Need to re-run the study so that the labels get changed to s_1 and s_2 (might actually have the data stored on the HPC already; if so, just use that).

In this section, we compare out-of-sample predictions from **FRK** version 2.0.0 to those from the R packages **INLA** version 20.03.17 (Lindgren and Rue 2015), **spNNGP** version 0.1.4 (Finley *et al.* 2020), **spBayes** version 0.4.3 (Finley *et al.* 2015), and **mgcv** version 1.8.33 (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 33,750 pixels, by creating a 150 × 225 grid and computing the average value of the response within each grid cell. Then, as the data provided by the MODIS instrument is continuous (measuring spectral radiances in units of $\mathrm{W/m^2/\mu m/st}$), we applied a reasonable threshold to obtain a binary version of the data: Specifically, pixels with spectral radiance greater than 7,000 were labelled "Cloud", and the remaining pixels were labelled "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 6,000 pixels for training, leaving 27,750 pixels for testing. The second sampling scheme, which we refer to as 'missing-in-a-block' (MB), involved excluding all pixels within a block for training, and using pixels inside the block for testing. The training block is a 30 × 30 square (900 pixels) in the middle of the spatial domain of interest, leaving 32,850 pixels for testing. 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

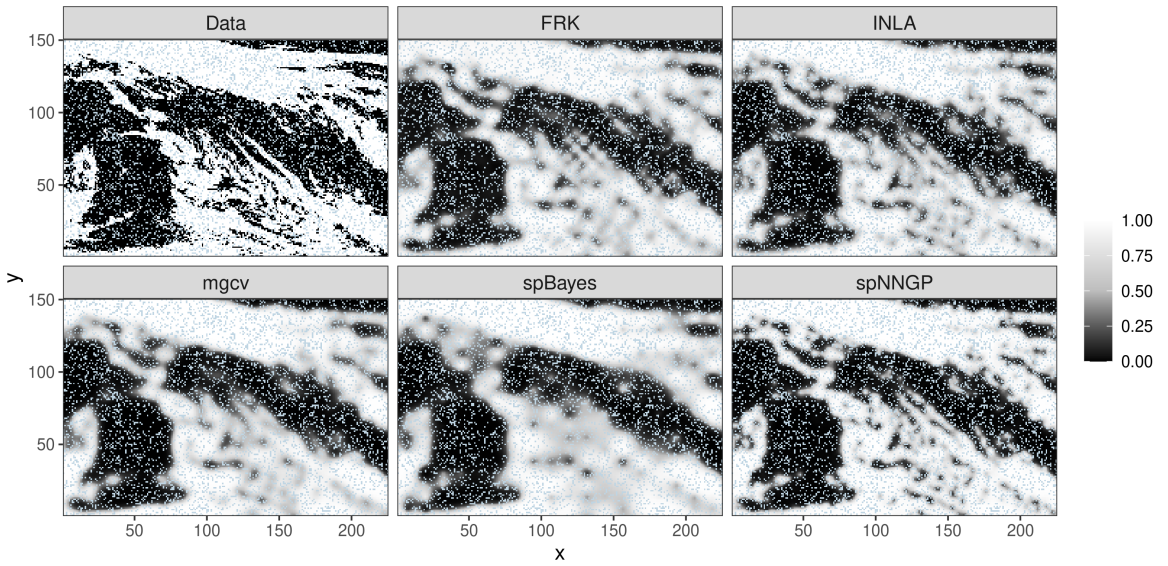

Figure 8: Predictions of the probability of 'Cloud' resulting from the missing-at-random data shown in Figure 7; the test data corresponding is shown in the top-left panel. Note that the training locations are indicated by blue pixels.

approach to model-selection by splitting the training data set in two, and then using one half for model fitting and the other half for model evaluation. In this way, we were able to evaluate a large number of arguments for each package and choose the best combination 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**, we used four resolutions of basis functions, giving a total of 11,130 basis functions. For **INLA**, we discretised the domain into 8,671 elements. For **mgcv**, we used the `bam()` function, which is similar to the generalised additive model function `gam()` but optimised for large data sets, with 3,000 knots. For **spBayes**, we used 400 knots; increasing the number of knots further was computationally prohibitive. When using **spNNGP**, we found that the default option of considering 15 neighbours at a time was appropriate. The packages **spNNGP** and **spBayes** use Markov chain Monte Carlo (MCMC): At both training and test locations, we used 10,000 total MCMC samples, with a burn-in of 6,000 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 (a known issue, at the time of writing, 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 from **FRK** v2, **INLA**, and **spNNGP** are similar, while the predictions from **mgcv** are slightly smoother than those from the aforementioned packages. The predictions of **spBayes** are even smoother, which is due to the small number of knots. Figure 9 shows the predictions resulting from the MB data shown in Figure 7. The packages **FRK** v2 and **INLA** return predictive probabilities close to 0.5, while **mgcv** and **spBayes** are more confident in their

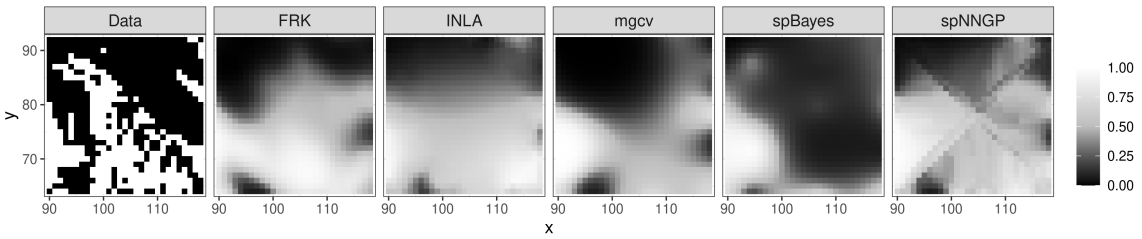

Figure 9: Predictions of the probability of 'Cloud' resulting from the 'missing-in-a-block' data shown in Figure 7. Here, we have shown only the testing locations, which corresponds to the 30×30 block near the centre of the spatial domain; the test data corresponding to this block is shown in the left-most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

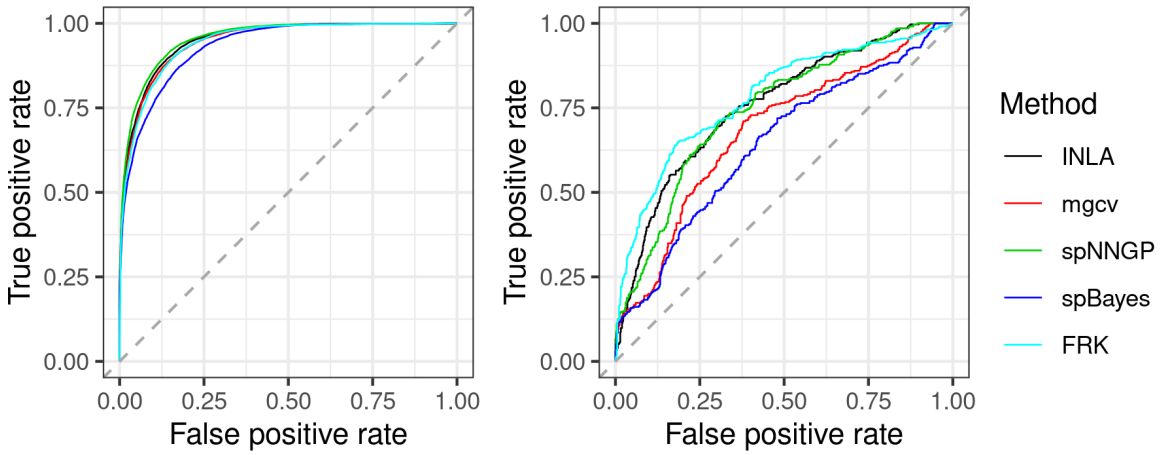

Figure 10: ROC curves for the training/test sets displayed in Figure 7. (Left panel) ROC curves generated from the 'missing-at-random' data. Note that there is some degree of overlap between **FRK**, **INLA**, and **spNNGP**. (Right panel) ROC curves generated from the 'missing-in-a-block' data.

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, $\pi(\cdot)$. However, the underlying distribution of the probability process is unidentifiable ?, as the 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. Noel doesn't follow this paragraph. Could explain it better if we had a formula for the binomial distribution.

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: 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 v2	0.09	0.95	**9.78**
	INLA	0.09	0.95	54.44
	mgcv	0.09	0.95	45.77
	spBayes	0.11	0.93	68.01
	spNNGP	**0.08**	**0.96**	12.35
MB	FRK v2	**0.19**	**0.78**	**17.74**
	INLA	0.20	0.76	141.85
	mgcv	0.23	0.69	186.67
	spBayes	0.25	0.63	489.41
	spNNGP	0.20	0.75	59.30

Table 5, while the ROC curves are shown in Figure 10. For the MR sampling scheme, there is little discernible difference between **FRK** v2, **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 sampling scheme are significantly worse than for the MR sample scheme. In this case, we see **FRK** v2, **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 increases in the run times for other packages 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 6,000 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 or better than other packages in this application. The main advantages of **FRK** v2 however lie in the ease with which it allows one to do other more elaborate analyses with spatial or spatio-temporal non-Gaussian data, as shown in the next sections.

4.2. Block prediction: Contaminated soil

Between the years 1954 and 1963, nuclear devices were detonated at Area 13 of the Nevada Test Site in the United States, contaminating the surrounding soil with the radioactive element americium (Am). The data set we use in this example is comprised of Am concentrations (in 10^3 counts per minute) in a spatial domain immediately surrounding Ground Zero (GZ; the location where the devices were detonated); it was previously analysed by Huang, Yao, Cressie, and Hsing (2009) and Paul and Cressie (2011). The total number of measurements (including some that are co-located) is 212. The left and centre panels of Figure 11 show the data on the original scale and on the log scale, respectively. Paul and Cressie (2011) note that the Am concentrations are clearly lognormally distributed, and that soil remediation is often done by averaging the contaminant over pre-specified spatial regions of D called blocks. Hence, this application requires lognormal prediction over blocks, a task well suited to **FRK** v2. The right panel of Figure 11 shows two blocking schemes that we predict over: Both

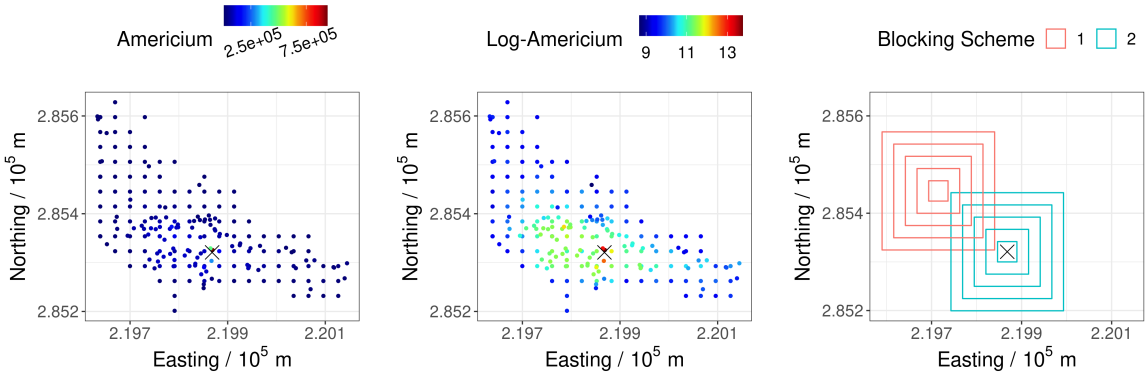

Figure 11: Americium soil data. The '×' denotes Ground Zero (GZ), where the devices were detonated. (Left panel) Am concentrations on the original scale. (Centre panel) Am concentrations on the log scale. (Right panel) Blocking schemes: Scheme 1 (red), centred away from GZ, and Scheme 2 (blue), centred on GZ; for context, the black points show the measurement locations. When the data locations overlap, we show only one which one? measurement (Noel is confused about this sentence). In the right-panel, red and blue has nothing to do with the colour scale used in the left and centre-left panels; possibly use another aesthetic, or completely different colours (e.g., purple and green)?

schemes contain 5 blocks, but one scheme is centred away from GZ, and the other is centred on GZ.

As in Paul and Cressie (2011), we use a piecewise linear trend, where observations within a distance of 30.48m (100 ft) from GZ are assumed to follow a different trend to those observations beyond 30.48m from GZ. The following code constructs the BAU-level covariates that are needed to fit this piecewise linear trend: `BAUs$x1` and `BAUs$x3` are indicator variables used to model the intercepts in each region, and `BAUs$x2` and `BAUs$x4` are used to model the slopes in each region.

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

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. In the formula below, we suppress the global intercept since we model the region-specific intercepts separately through `BAUs$x1` and `BAUs$x3`.

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

In order to generate block-level predictions, we pass a '`SpatialPolygonsDataFrame`' object into the `newdata` argument of `predict()`. In the following code, `blocks` is a

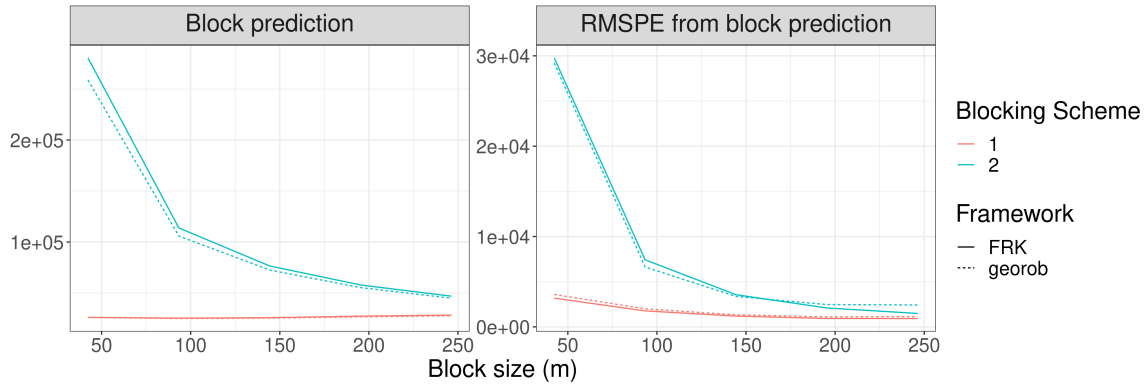

Figure 12: Block-predictions (left panel) and root mean squared prediction error (right panel) of Am concentrations against block size, $|B|^{1/2}$. Both quantities are in units of 10^3 counts per minute. In both panels, the red line corresponds to Scheme 1 and the blue line corresponds to Scheme 2.

'`SpatialPolygonsDataFrame`' object containing the polygons corresponding to the two blocking schemes shown in Figure 11:

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

To validate 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). (Noel says that the paper gives an *exact* unbiased back-transformation; need to check to see what is going on. Could just remove "approximately unbiased" and the reference to Cressie (2006).) Computation times for kriging do not scale well for large sample sizes, however the size of this data set is sufficiently small for straightforward kriging to be possible. The package **georob** provides users with two approaches to lognormal block kriging; here, 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. It can be clearly seen that the results corroborate each other, and are practically identical despite the use of dimension reduction in **FRK** v2.

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

The Australian Statistical Geography Standard (ASGS) defines a series of nested geographical areas in Australia known as Statistical Area Levels. Statistical Area Level 3 (SA3) regions are aggregations of Statistical Area Level 2 (SA2) regions, and SA2 regions are aggregations of Statistical Area Level 1 (SA1) regions. In this example, we consider a region of the state of New South Wales in Australia, which contains 7,909 SA1 regions, 180 SA2 regions, and 31 SA3 regions, and we aim to infer 'poverty' levels at the SA1 and SA3 regions just from a data set containing mostly SA2 data and a small amount of SA1 data. The data were collected in the Census of 2011, and they consist of the number of families of various types within a range of weekly income brackets; in Appendix D, we provide further details on the way in which we define the poverty line for each family type. 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.

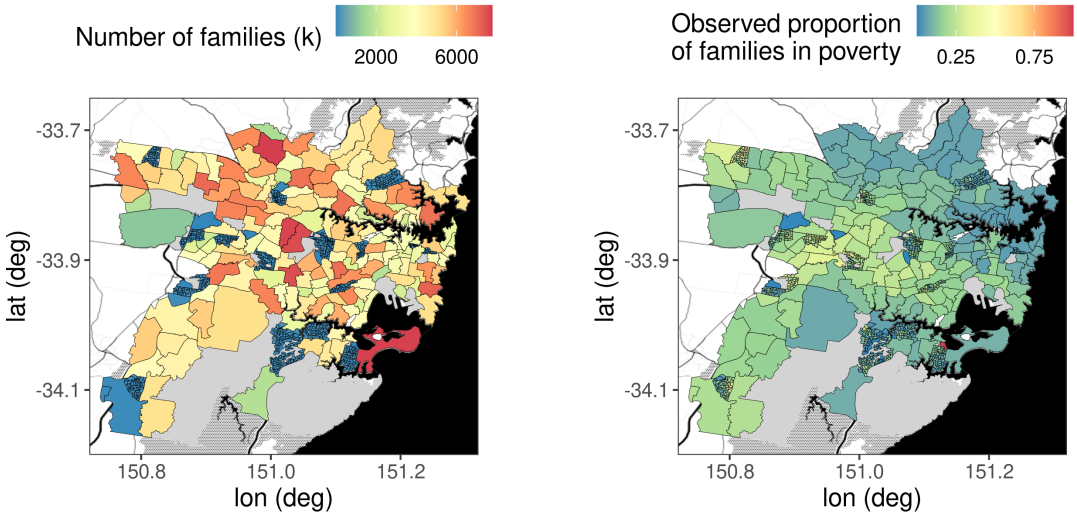

Figure 13: Training data on SA1/SA2 regions used for modelling the number (or proportion) of families 'in poverty' (see Appendix D for how we define 'in poverty'). (Left panel) The total number of families. (Right panel) The observed proportion of families in poverty, computed by dividing the number of families in poverty by the total number of families. Solid-grey regions correspond to SA1/SA2 regions in which the total number of families is zero. The data are overlayed on a Stamen base map, where the textured grey areas correspond to bushland, large black areas correspond to ocean, and lines correspond to major arterial roads. Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

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-region (and some SA1-region) data for training the model, and we use 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 over the SA1 regions. Need to give the data (binomial) model and link function in the text (as a formula).

```
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 all of the SA1 regions.

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

Since the SA1 regions are our BAUs, we can predict both the probability process and the mean process over the SA1 regions: We focus on the probability process, which is independent of the size parameter. (Give the data model on the previous page, where the prob process

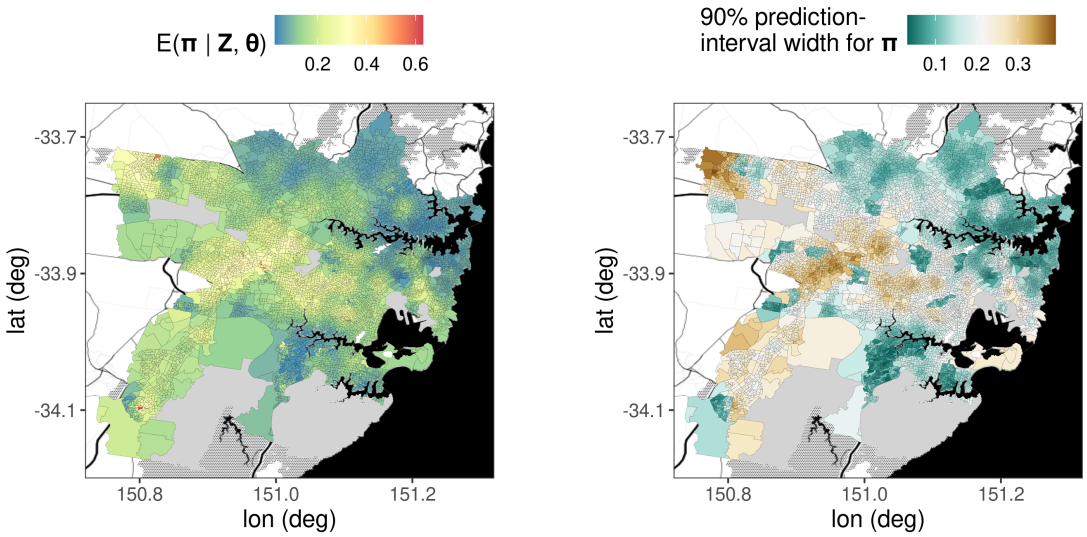

Figure 14: SA1-level predictions. (Left panel) Prediction of the probability process, $\pi(\cdot)$, representing the proportion of families in poverty, over the SA1 regions. (Right panel) 90% prediction-interval width of the probability process. Solid-grey regions correspond to SA2 regions in which the total number of families is zero, and hence they are omitted from the study. For details on the underlying Stamen base map, refer to the caption of Figure 13.

and the size parameter are defined.) 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 prediction-interval widths of the mean process. Again, this graphic was generated using `plot()`.

We assessed the model's ability to quantify uncertainty over the SA1 regions by computing the empirical coverage from nominal 90% prediction intervals. We found the empirical coverage to be 90.8%, which 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 automatically fixes σ^2_ξ before fitting the model 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 (e.g., 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 implemented over any arbitrary, user-specified polygons; see Section 3.2 for an illustration.

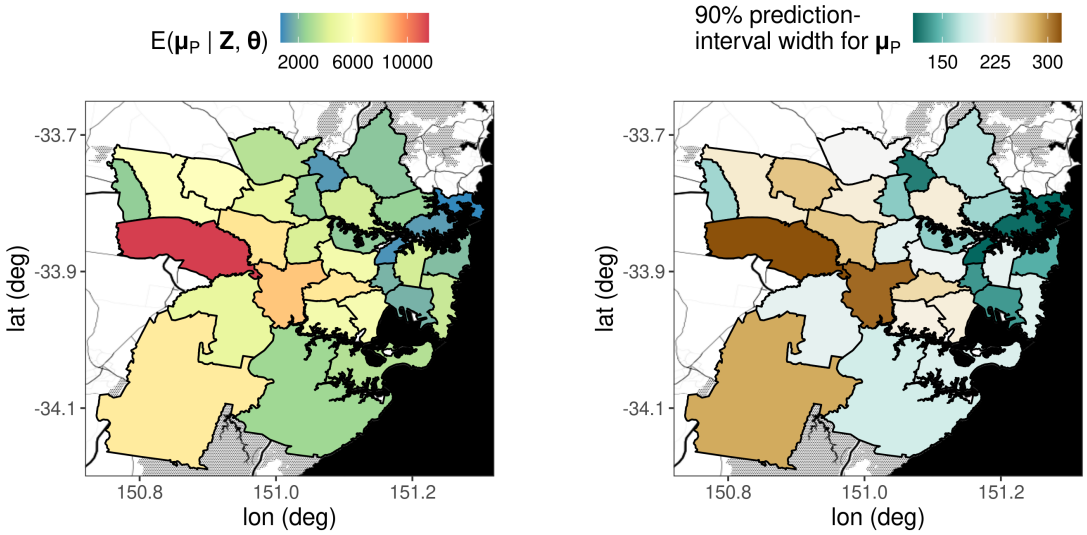

Figure 15: SA3-level predictions. (Left panel) Prediction of the mean process, $\mu(\cdot)$, representing the expected number of families in poverty, over the SA3 regions. (Right panel) The 90% prediction-interval width of the mean process. For details on the underlying Stamen base map, refer to the caption of Figure 13.

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 1920s (The University of Chicago Library 2020). In this study, we model the number of crimes in each CA between the years 2001 and 2019 inclusive. A full list of crimes committed in Chicago during this period is provided by the Chicago Police Department and is available for download from the open-data-source website Plenario (Urban Center for Computation and Data of the University of Chicago 2020). We considered only crimes labelled as assault or battery; there were roughly 1.75 million crimes in total. Note that the default behaviour of all releases of **FRK** is to bin data falling into the same BAU; in this example, the final number of observations post-binning is 1444. Individual-level data and aggregated-level data 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()`. Recall that, in this case, the spatio-temporal BAUs are space-time volumes constructed by taking all combinations of the spatial BAU footprints with the yearly intervals that make up the 19-year period of interest.

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

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

This covariate was obtained from the 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 was constant over the time-span of the data.

```
R> ST_BAUs$population <- community_areas$population
```

Next, we generate spatio-temporal basis functions automatically using `auto_basis()`.

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

Then, we initialise and fit the 'SRE' object using `FRK()`, setting `response = "poisson"` and `link = "log"`. (refer back to the Poisson model that we will define earlier in the paper.) Each entry in our data set 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 counting the number of crimes when 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 taking an average. To do so, we pass the name of the 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 to each spatial BAU its own fine-scale variance parameter by setting `fs_by_spatial_BAU = TRUE` (see Section 2.5). To validate predictions, we excluded the years 2010 and 2019 from the training data, and used them to evaluate crime predictions and forecasts, respectively.

```
R> S <- FRK(f = number_of_crimes ~ log(population),
+    data = 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 and individual years as temporal BAUs) using `predict()`, and plot the results using `plot()`.

The true (withheld) number of crimes, predicted number of crimes, and prediction uncertainty for the prediction (2010) and forecast (2019) years are shown in Figure 16. For both years, Figure 16 shows agreement between the predicted and actual number of crimes. Furthermore, the prediction uncertainty is roughly proportional to the predicted value, as expected when counts are modelled. For the prediction year and forecast year, we also computed the empirical coverage when using 90%, 80%, 70%, and 60% prediction intervals, and the mean absolute percentage error (MAPE; see Appendix C). We consistently observed that the empirical coverage in the year 2010 was slightly (4% on average) higher than the nominal coverage, whilst it was lower (13% on average) in the forecast year. We observed MAPE scores of 4% and 9% in the years 2010 and 2019, respectively. The slightly worse results in the year 2019 is expected, as forecasting into the future is a harder task than predicting within the time span of the data. Nonetheless, these predictions are cause for optimism given the complexity of modelling crime in a spatio-temporal setting.

Next, we next focus on three randomly-selected CAs: Ashburn, Roseland, and Archer Heights. The time series of the observed data, predictions, and 90% prediction intervals for these

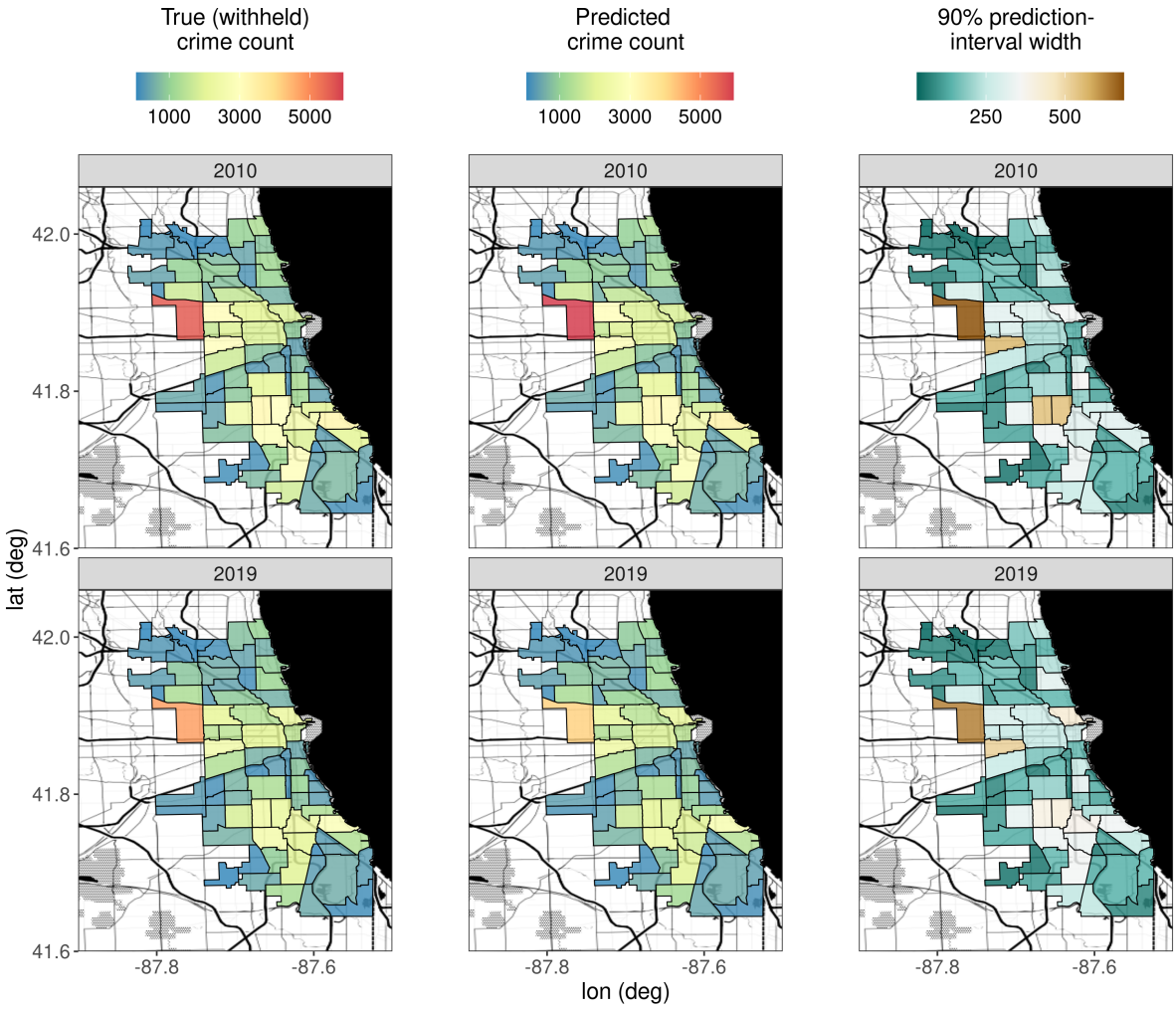

Figure 16: True (withheld) number of crimes, predictions, and prediction-interval width over Chicago in the prediction (2010) and forecast (2019) years. The first row corresponds to the year 2010, and the second row corresponds to the year 2019. The first column shows the true (withheld) number of crimes; the second column shows the predicted number of crimes; and the third column shows the width of a prediction interval with a nominal coverage of 90%. For details on the underlying Stamen base map, refer to the caption of Figure 13 (the large black regions in this figure correspond to Lake Michigan).

CAs, are shown in Figure 17. The prediction intervals are slightly wider in validation years (2010 and 2019) than in observed years. The observed number of crimes is contained within the prediction 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 Ashburn and Roseland in the year 2019 are particularly accurate, with the predicted/forecasted number of crimes essentially equal to the true (withheld) number of crimes.

5. Conclusion

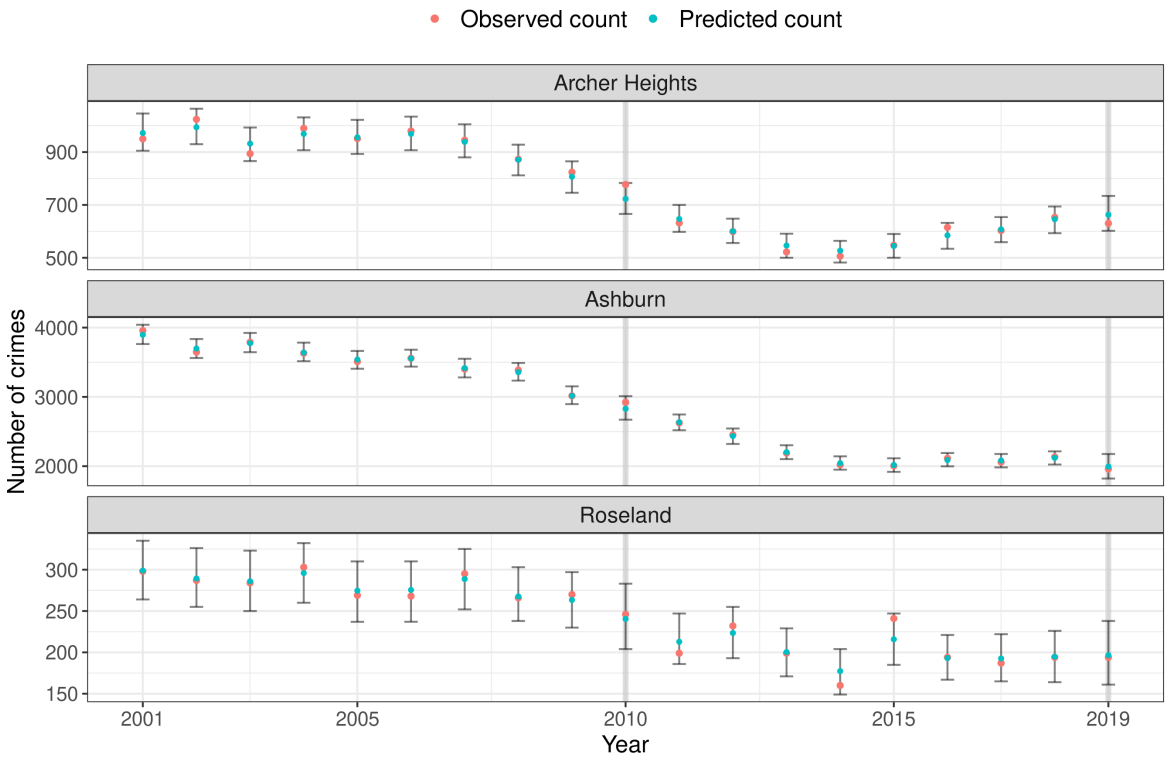

Figure 17: Time-series plots of predicted and observed number of crimes for three randomly-selected CAs. The prediction (2010) and forecast (2019) years are highlighted in light-grey. The observed number of crimes at each time is indicated by a red dot, whilst the predicted number of crimes is indicated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

In this paper, we have described **FRK** v2, a major extension to the R package **FRK**, referred to as **FRK** v1. Substantial enhancements allow for the spatial and spatio-temporal modelling of, and large-scale prediction from, big, non-Gaussian data. Using a GLMM model and the software **TMB**, **FRK** v2 can now cater for many distributions within the exponential family, as well as many link functions. Furthermore, **FRK** v2 allows for the use of many more basis functions when modelling the spatial process, and hence it can often achieve more accurate predictions in a Gaussian setting than **FRK** v1. The existing functionality of **FRK** v1 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 facilitates 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 requirement of the framework is that covariates need to be known for every BAU, which may not be the case if covariates are recorded only at the data-support level. Spatial interpolation of the covariates can be used to address this problem. Another requirement is

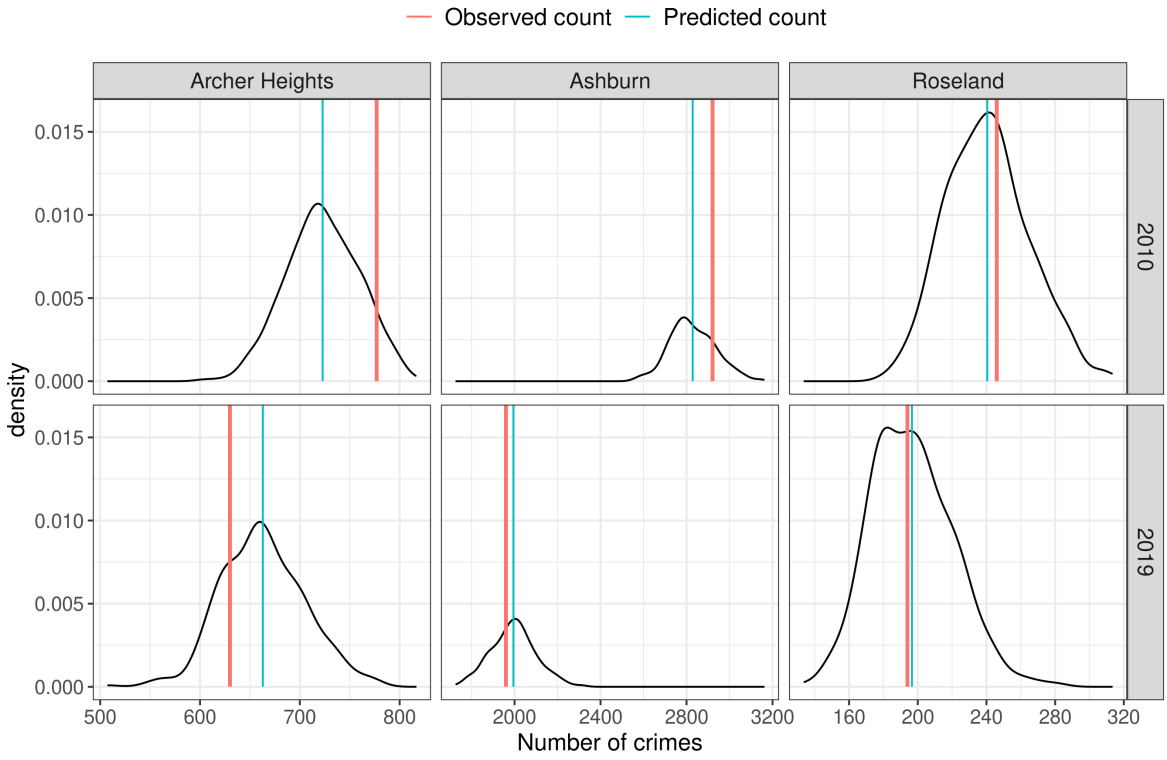

Figure 18: Predictive distributions in the prediction (2010) and forecast (2019) years for three randomly-selected CAs (Archer Heights, Ashburn, and Roseland). In each panel, the red line corresponds to the true (withheld) number of crimes and the blue line corresponds to the predicted number of crimes. The first row corresponds to the year 2010 and the second row corresponds to the year 2019. The first, second, and third columns correspond to Archer Heights, Ashburn, and Roseland, respectively.

the practical necessity to fix the fine-scale variance parameter in spatial change-of-support applications; note that this is not an issue if one is able to obtain a reliable estimate through other means (e.g., via previously sampled data or via quality-control experiments). We are currently exploring avenues to relax this requirement via the provision of a robust offline estimate. A limitation of **FRK** v2 is that, despite the added flexibility, several models of interest, such as the zero-inflated Poisson, are still not catered for; future work will see the introduction of other models of interest. 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. The main spatial data classes used in **FRK** v2 come from the package **sp** (Pebesma and Bivand 2005); future work may add support for other spatial data classes, such as those defined in 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 discussions 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, New York, NY.

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: Series B*, **70**, 209–226.

Cressie N, Sainsbury-Dale M, Zammit-Mangion A (2021). "Basis-function models in spatial statistics." *Annual Review of Statistics and its Applications*, **In press**.

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: Series C*, **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**(13), 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: Series B*, **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 JR (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." *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**(5), 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**(19), 1–25.

Lindgren F, Rue H, Lindström J (2011). "An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach." *Journal of the Royal Statistical Society: 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: Series B*, **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 of the 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, NY. 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*, **98**(4), 1–48.

Zammit-Mangion A, Ng TLG, Vu Q, Filippone M (2021). "Deep compositional spatial models." *Journal of the American Statistical Association*. doi: 10.1080/01621459.2021.1887741.

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

A. Parametrisations of the basis-function coefficients

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

A.1. Covariance matrix $\boldsymbol{K}$

Let $K_k(\boldsymbol{s}, \boldsymbol{s}^*)$ denote the covariance function associated with the basis-function coefficients corresponding to the kth basis-function resolution. In **FRK**, we let $K_k(\boldsymbol{s}, \boldsymbol{s}^*)$ be 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 two basis-function centroids $\boldsymbol{s}, \boldsymbol{s}^* \in D$. The kth sub-block of $\boldsymbol{K}$ is formed by evaluating A.1 for all pairs of basis-function centroids at the kth resolution.

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 now 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 Matérn smoothness parameter $\nu = 0.5$, 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 let β_k be proportional to 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 $\boldsymbol{Q}$

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

When the basis functions are regularly spaced (`regular = TRUE`), **FRK** v2 uses a precision matrix that is related to that used in the R package **LatticeKrig** (Nychka, Hammerling, Sain, and Lenssen 2016). Let $\mathcal{N}_{ik}$ 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 precision matrix of the basis-function coefficients at resolution k.

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

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

where κ_k and ρ_k are parameters that are 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. Note that **LatticeKrig** uses $\boldsymbol{Q}_k^\top \boldsymbol{Q}_k$ as the precision matrix blocks, whereas **FRK** v2 uses $\boldsymbol{Q}_k$. LeRoux model?

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

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

where κ_k, ρ_k, and κ_k are parameters that are estimated, and $T_{\beta_k}(\cdot, \cdot)$ is defined as in Appendix A.1. Again, this matrix is diagonally dominant and hence is 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

(Refer to the data models introduced earlier in the paper.)

Two data models that can be used with **FRK** v2, namely, the binomial distribution and the negative-binomial distribution, 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 (i.e., a function that 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 inverse link function, $g^{-1}(\cdot)$, is $(0, 1)$. However, the mean of a 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

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

Using a hierarchical-link-function approach with a negative-binomial data model and $f(\cdot)$ a logit function, we therefore 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

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

Observe that $\pi(\cdot) \in (0, 1)$, so that $\mu(\cdot) \in (0, \infty)$, which is an appropriate range for modelling the expectation of a negative-binomial distributed random variable, $Z(\cdot) \sim \text{NB}(\mu(\cdot); k(\cdot))$. Now consider a binomial data model, $Z(\cdot) \sim \text{Bin}(\mu(\cdot); k(\cdot))$, where we use the notation $k(\cdot)$ to emphasise that the size parameter is related to the spatial support of the data. In this case, we have that

$$h(\mu(\cdot); k) = \frac{\mu(\cdot)}{k} = \pi(\cdot),$$

and since $\pi(\cdot) \in (0, 1)$, we have $\mu(\cdot) \in (0, k)$, which is an appropriate range for modelling the expectation of a binomial distributed random variable.

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 via $h(\cdot)$ using the known form of the mean, specific to the distribution in question. 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 diagnostics (for simplicity, we describe the diagnostics in terms of prediction of the continuous mean process):

- (Empirical) root-mean-squared prediction error (RMSPE): Let $\hat{\mu}(\boldsymbol{s})$ 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 (empirical) 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 (empirical) mean-absolute percentage error, is similar to the MAE, but here 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 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

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

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

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

D. Sydney poverty lines

Here we provide some details on how we define the poverty lines for the data in Section 4.3. We base our definitions of poverty lines on a Melbourne Institute of Applied Economic and Social Research (MIAESR) report that was published in March 2011 (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 we use the first half of Table 1 of the MIAESR report guidelines. Fourth, our data do not provide exact income figures, but rather they provide income brackets of width \$200; we thus round MIAESR guidelines to the nearest \$200. These assumptions lead us to define poverty lines (in Australian dollars) for each family unit considered in this study as weekly incomes of: \$600 for a couple with no children, \$800 for a couple with children, and \$600 for a one-parent family. The proportion of families we deem to be in poverty in each region is based on these thresholds.

Affiliation:

Matthew Sainsbury-Dale
National Institute for Applied Statistics Research Australia (NIASRA)
School of Mathematics and Applied Statistics
University of Wollongong
Wollongong, Australia
E-mail: msainsburydale@gmail.com
URL: https://github.com/MattSainsbury-Dale

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd