

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

Matthew Sainsbury-Dale, Andrew Zammit-Mangion, and Noel Cressie

May 6, 2021

## Abstract

Non-Gaussian spatial and spatial-temporal data are becoming increasingly prevalent, arising from studies as far apart as small area demographics (counts) and global remote sensing (radiant energies). **FRK** is an R package for spatial/spatio-temporal modelling and prediction with large data sets. In this vignette, we demonstrate the modelling of non-Gaussian data using **FRK**. The existing functionality of **FRK** is retained with this advance; in particular, it makes use of automatic basis-function construction, it can handle both point-referenced and areal data simultaneously, and it predicts process values at any spatial support from these data. See the vignette “FRK\_intro” for general usage of the package.

## Contents

<b>1</b>	<b>Methodology</b>	<b>1</b>
1.1	The process layer . . . . .	1
1.2	The data layer . . . . .	2
1.3	Estimation . . . . .	3
1.4	Prediction and uncertainty quantification . . . . .	3
1.4.1	Arbitrary prediction regions . . . . .	4
<b>2</b>	<b>Example: Non-Gaussian, point-referenced spatial data</b>	<b>4</b>

## 1 Methodology

The statistical model used by **FRK** in a non-Gaussian setting is a spatial generalised linear mixed (GLMM) model [Diggle et al., 1998], a hierarchical model consisting of two layers. In the *process* layer, we model the conditional mean of the data as a transformation of a latent spatial process, where the spatial process is modelled as a low-rank spatial random effects model; see Section 1.1. The process layer, which governs the conditional mean of the data, retains many similarities to that in previous versions of the package, as described by Zammit-Mangion and Cressie [2021] and in the vignette “FRK\_intro”. In the *data* layer, we use a conditionally independent exponential-family model for the data; see Section 1.2. In Sections 1.3 and 1.4 we briefly discuss parameter estimation, and spatial prediction and uncertainty quantification.

### 1.1 The process layer

Denote the latent spatial process as  $Y(\cdot) \equiv \{Y(\mathbf{s}) : \mathbf{s} \in D\}$ , where  $\mathbf{s}$  indexes space in the spatial domain of interest  $D$ . The model for  $Y(\cdot)$  is the so-called spatial random effects (SRE) model [Cressie and Johannesson, 2008],

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

where  $\mathbf{t}(\cdot)$  are spatially referenced covariates with associated regression parameters  $\boldsymbol{\alpha}$ ,  $\boldsymbol{\phi}(\cdot) \equiv (\phi_1(\cdot), \dots, \phi_r(\cdot))^\top$  is an  $r$ -dimensional vector of pre-specified spatial basis functions with associated random coefficients  $\boldsymbol{\eta}$ , and  $\xi(\cdot)$  is a fine-scale random process that is ‘almost’ uncorrelated.

**FRK** discretises the domain of interest  $D$  into  $N$  small, non-overlapping basic areal units (BAUs)  $\{A_i : i = 1, \dots, N\}$  such that  $D = \cup_{i=1}^N A_i$ . BAUs are a key element of **FRK**, as they provide a framework that allows one to use both point-referenced and areal data simultaneously, and one that facilitates the spatial change-of-support problem. After discretisation via the BAUs, we obtain the vectorised version of (1),

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

where  $\mathbf{Y}$  is an  $N$ -dimensional vector,  $\mathbf{T}$  and  $\mathbf{S}$  are known design matrices, and  $\boldsymbol{\xi}$  is the vector associated with the fine-scale process.

We model the fine-scale random effects as being independent and identically distributed Gaussian random variables with variance  $\sigma_\xi^2$ ; we model  $\boldsymbol{\eta}$  as a mean-zero multivariate-Gaussian random variable, typically using a sparse precision matrix parametrisation formulation in a non-Gaussian setting.

Following standard generalised linear model theory [McCullagh and Nelder, 1989], **FRK** v.2 uses a link function,  $g(\cdot)$ , to model  $Y(\cdot)$  as a transformation of the mean process,  $\mu(\cdot)$ :

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

The mean process evaluated over the BAUs is

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

where  $g^{-1}(\cdot)$  is the inverse link function. An identity link function and a Gaussian data model yields the standard Gaussian FRK model.

## 1.2 The data layer

Given  $m$  observations with footprints spanning one or more BAUs, we define the observation supports as  $B_j \equiv \cup_{i \in c_j} A_i$  for  $j = 1, \dots, m$ , where  $c_j$  is a non-empty set in the power set of  $\{1, \dots, N\}$ , and define  $D^O \equiv \{B_j : j = 1, \dots, m\}$ . Let  $Z_j \equiv Z(B_j)$ ,  $j = 1, \dots, m$ . The vector of observations (the data vector) is then  $\mathbf{Z} \equiv (Z_1, \dots, Z_m)^\top$ .

Since each  $B_j \in D^O$  is either a BAU or a union of BAUs, one can construct an  $m \times N$  matrix

$$\mathbf{C}_Z \equiv \left( w_i \mathbb{I}(A_i \subset B_j) : i = 1, \dots, N; j = 1, \dots, m \right),$$

where  $\mathbb{I}(\cdot)$  is the indicator function, which creates a linear mapping from  $\boldsymbol{\mu} \equiv (\mu_i : i = 1, \dots, N)^\top$  to evaluations of the mean process over the observation supports;

$$\boldsymbol{\mu}_Z \equiv \mathbf{C}_Z \boldsymbol{\mu}. \quad (3)$$

The `normalise_wts` argument in `SRE()` controls whether the linear mapping of  $\mathbf{C}_Z$  corresponds to a weighted sum or a weighted average; if `normalise_wts = TRUE`, then the weights  $w_i$  are normalised so that the rows of  $\mathbf{C}_Z$  sum to one, and the mapping represents a weighted average.

We assume that  $[Z_j \mid \mu(\cdot), \psi] = [Z_j \mid \boldsymbol{\mu}_{Z,j}, \psi]$ , where  $\psi$  is a (nuisance) dispersion parameter and, for a generic random quantities  $A$  and  $B$ ,  $[A \mid B]$  denotes the probability distribution of  $A$  given  $B$ . That is, a given observation depends only on the value of the mean process at the corresponding observation support, rather than on the process over the whole domain. As a result, conditional on the latent spatial process, all observations are conditionally independent:

$$[\mathbf{Z} \mid \mu(\cdot), \psi] = \prod_{j=1}^m [Z_j \mid \boldsymbol{\mu}_{Z,j}, \psi].$$

We model the conditional distribution  $[Z_j \mid \boldsymbol{\mu}_{Z,j}, \psi]$  as a member of the exponential family [McCullagh and Nelder, 1989, Sec. 2.2.2], with conditional expectation  $\mu(B_j) \equiv \mathbb{E}\{Z_j \mid \boldsymbol{\mu}_{Z,j}, \psi\}$ .

The model employed by **FRK** v.2 can be summarised as follows.

$$Z_j \mid \boldsymbol{\mu}_{Z,j}, \psi \stackrel{\text{ind}}{\sim} \text{EF}(\boldsymbol{\mu}_{Z,j}, \psi), \quad j = 1, \dots, m, \quad (4)$$

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

$$g(\boldsymbol{\mu}) = \mathbf{Y}, \quad (6)$$

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\alpha} + \mathbf{S}\boldsymbol{\eta} + \boldsymbol{\xi}, \quad (7)$$

$$\boldsymbol{\eta} \mid \boldsymbol{\vartheta} \sim \text{Gau}(\mathbf{0}, \mathbf{Q}^{-1}), \quad (8)$$

$$\boldsymbol{\xi} \mid \sigma_\xi^2 \sim \text{Gau}(\mathbf{0}, \sigma_\xi^2 \mathbf{V}). \quad (9)$$

where  $\mathbf{V}$  is a known diagonal matrix with positive entries on the diagonal and  $\sigma_\xi^2$  is either unknown and estimated, or provided by the user.

### 1.3 Estimation

The complete-data likelihood function for our model is

$$L(\boldsymbol{\theta}; \mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}) \equiv [\mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}] = [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi][\boldsymbol{\eta} \mid \boldsymbol{\vartheta}][\boldsymbol{\xi} \mid \sigma_\xi^2], \quad (10)$$

where  $\boldsymbol{\theta} \equiv (\boldsymbol{\alpha}^\top, \boldsymbol{\vartheta}^\top, \sigma_\xi^2, \psi)^\top$  and  $\boldsymbol{\vartheta}$  are variance components, and its logarithm is

$$l(\boldsymbol{\theta}; \mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}) \equiv \ln L(\boldsymbol{\theta}; \mathbf{Z}, \boldsymbol{\eta}, \boldsymbol{\xi}) = \ln [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi] + \ln [\boldsymbol{\eta} \mid \boldsymbol{\vartheta}] + \ln [\boldsymbol{\xi} \mid \sigma_\xi^2]. \quad (11)$$

Under our modelling assumptions, the conditional density functions  $[\boldsymbol{\eta} \mid \boldsymbol{\vartheta}]$  and  $[\boldsymbol{\xi} \mid \sigma_\xi^2]$  are invariant to the specified link function and assumed distribution of the response variable. Of course, this invariance does not hold for  $[\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi]$ . As we only consider data models in the exponential family,  $\ln [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi]$  may be expressed as

$$\ln [\mathbf{Z} \mid \boldsymbol{\mu}_Z, \psi] = \sum_{j=1}^m \left\{ \frac{Z_j \lambda(\boldsymbol{\mu}_{Z,j}) - b(\lambda(\boldsymbol{\mu}_{Z,j}))}{a(\psi)} + c(Z_j, \psi) \right\}, \quad (12)$$

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

The marginal likelihood, which does not depend on the random effects, is given by

$$L^*(\boldsymbol{\theta}; \mathbf{Z}) \equiv \int_{\mathbb{R}^p} L(\boldsymbol{\theta}; \mathbf{Z}, \mathbf{u}) d\mathbf{u}, \quad (13)$$

where  $\mathbf{u} \equiv (\boldsymbol{\eta}^\top, \boldsymbol{\xi}^\top)^\top \in \mathbb{R}^p$ , and  $p$  is the total number of random effects in the model. When the data are non-Gaussian, the integral in (13) is typically intractable and must be approximated either numerically or analytically. In **FRK**, we use the Laplace approximation, implemented using the R package **TMB** [Kristensen et al., 2016].

Given as input a C++ template function which defines the complete-data log-likelihood function (11), **TMB** [Kristensen et al., 2016] computes the Laplace approximation of the marginal log-likelihood, and automatically computes its derivatives, which are then called from within **FRK** by an optimising function specified by the user (`nlminb()` is used by default). **TMB** uses **CppAD** [Bell, 2005] for automatic differentiation, and the linear algebra libraries **Eigen** [Guennebaud et al., 2010] and **Matrix** [Bates et al., 2019] for vector and matrix operations in C++ and R, respectively; use of these packages yields good computational efficiency. **TMB**'s implementation of automatic differentiation is a key reason why we can cater for a variety of response distributions and link functions, as we do not need to consider each combination on a case-by-case basis.

### 1.4 Prediction and uncertainty quantification

There are three primary quantities of interest in this framework: The latent process  $Y(\cdot)$ , the mean process  $\boldsymbol{\mu}(\cdot)$ , and the noisy data process. To produce predictions and associated uncertainties, we need to determine the posterior distribution of these quantities.

It can be shown that the Laplace approximation implies that the posterior distribution of the random effects,  $\mathbf{u} \mid \mathbf{Z}, \boldsymbol{\theta}$  is approximated to be Gaussian. This, in turn, implies that the posterior distribution of  $\mathbf{Y}$  is also approximated to be Gaussian, and hence inference on  $Y(\cdot)$  can be done using closed form solutions. However, the posterior distribution of non-linear functions of  $Y(\cdot)$  (e.g., the mean process) are typically not available in closed form, and in this case some form of approximation is required. Hence, we choose to use a Monte Carlo (MC) framework.

For each quantity, we use the posterior expectation as our predictor. A commonly used metric for uncertainty quantification is the root-mean-squared prediction error (RMSPE). In a non-Gaussian setting, it can be difficult to interpret the RMSPE, and it is often more intuitive to quantify uncertainty through the width of the posterior predictive intervals. Hence, in **FRK**, we also provide the user with user-specified percentiles of the posterior predictive distribution. These quantities can be computed straightforwardly using MC sampling.

#### 1.4.1 Arbitrary prediction regions

Often, one does not wish to predict over a single BAU, but over regions spanning multiple BAUs. Define the set of prediction regions as  $D^P \equiv \{\tilde{B}_k : k = 1, \dots, N_P\}$ , where  $\tilde{B}_k \equiv \cup_{i \in c_k} A_i$ , and where  $c_k$  is some non-empty set in the power set of  $\{1, \dots, N\}$ . Like the data, the prediction regions  $\{\tilde{B}_k\}$  may overlap. In practice,  $\tilde{B}_k$  may not include entire BAUs; in this case, we assume that a prediction region contains a BAU if and only if there is at least some overlap between the BAU and the prediction region. Prediction over  $D^P$  requires some form of aggregation across relevant BAUs. Since in the non-Gaussian setting aggregation must be done on the original scale, we restrict prediction over arbitrary regions to the mean (or the noisy data process). Therefore, predictions of the latent process  $Y(\cdot)$  are not allowed over arbitrary prediction regions.

Consider the predictions  $\{\mu_P(\tilde{B}_k) : k = 1, \dots, N_P\}$ , where  $\mu_P(\cdot) \equiv \mu(\cdot \mid \mathbf{Z}, \boldsymbol{\theta})$ . These predictions are weighted sums of the predictions over the associated BAUs. Specifically,

$$\mu_{P,k} \equiv \mu_P(\tilde{B}_k) = \sum_{i=1}^N \tilde{w}_i \mathbb{I}(A_i \subset \tilde{B}_k) \mu_i; \quad i = 1, \dots, N; \quad k = 1, \dots, N_P; \quad \tilde{B}_k \in D^P,$$

where, in a similar fashion to the incidence matrix  $\mathbf{C}_Z$ , the weights  $\{\tilde{w}_i\}$  are optionally provided by the user in the `wts` field of the BAU object, and may be normalised if `normalise_wts = TRUE`. If `wts` is `NULL`, the BAUs are assumed to be equally weighted. Define  $\boldsymbol{\mu}_P \equiv (\mu_{P,k} : k = 1, \dots, N_P)^\top$ . Since each element in  $D^P$  is the union of subsets of  $D^G$ , one can construct a matrix,

$$\mathbf{C}_P \equiv (\tilde{w}_i : i = 1, \dots, N; k = 1, \dots, N_P),$$

such that  $\boldsymbol{\mu}_P = \mathbf{C}_P \boldsymbol{\mu}$ . Again, we use MC sampling to predict  $\boldsymbol{\mu}_P$ .

## 2 Example: Non-Gaussian, point-referenced spatial data

In this part of the vignette we apply **FRK** to the case when we have non-Gaussian, point-referenced spatial data. For simplicity, in this vignette we only consider inference on the plane; for other manifolds, such as the sphere, see the vignette “FRK\_intro”.

```
library("FRK")      # for carrying out FRK
library("sp")       # for defining points/polygons
library("dplyr")    # for easy data manipulation
library("ggplot2")  # for plotting
library("ggpubr")   # for arranging plots
```

First, we simulate a Poisson data set.

```

m <- 1000 # Sample size
RNGversion("3.6.0"); set.seed(1) # Fix seed
zdf <- data.frame(x = runif(m), y = runif(m)) # Generate random locs
zdf$Y <- sin(8 * zdf$x) + cos(8 * zdf$y) + 0.5 * rnorm(m) # Simulate latent process
zdf$z <- rpois(m, lambda = zdf$Y^2) # Simulate data
coordinates(zdf) = ~x+y # Turn into sp object

```

Now, we run **FRK**. There is an ‘expert’ way of using **FRK**, which involves using the functions `auto_BAUs()` and `auto_basis()` to automatically construct BAUs and basis functions from the data, `SRE()` to initialise the SRE model object and perform data preprocessing, and `SRE.fit()` to fit the model object. This ‘expert’ way is documented in the vignette “FRK\_intro”. Alternatively, there is a ‘simple’ way of using **FRK** that uses the high-level wrapper function `FRK()` that calls these functions under-the-hood; in this vignette, we will use the ‘simple’ way.

```

S <- FRK(f = z ~ 1, # Formula to FRK
        list(zdf), # All datasets are supplied in list
        nres = 1, # Reduce run-time using only 1 basis function resolution
        response = "poisson", # Poisson data model
        link = "square-root") # square-root link function
pred <- predict(S) # prediction stage

```

**FRK** includes two plotting methods, `plot()` and `plot_spatial_or_ST()`; the former takes an SRE object and the result of a call to `predict`, and returns a list of ‘ggplot’ objects containing the predictions and uncertainty quantification of those predictions, while the former is a general-purpose function for ‘Spatial\*DataFrame’ and ‘STFDF’ objects. See Figure 1.

```

plot_list <- plot(S, pred, zdf)
ggarrange(plot_list$z + labs(fill = "data"),
          plot_list$p_mu + labs(fill = "pred."),
          plot_list$interval90_mu + labs(fill = "pred.\nuncertainty"),
          nrow = 1, legend = "top")

```

## References

- Douglas Bates, Martin Maechler, and Timothy A. Davis. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2019. URL <http://Matrix.R-forge.R-project.org/>. R package version 1.2-17.
- B. M. Bell. CppAD: a package for C++ algorithmic differentiation. <http://www.coin-or.org/CppAD>, 2005. Accessed: 2019-06-15.
- N. Cressie and G. Johannesson. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society*, 70:209–226, 2008.
- P. J. Diggle, J. A. Tawn, and R. A. Moyeed. Model-based geostatistics. *Journal of the Royal Statistical Society*, 47:299–350, 1998.
- G. Guennebaud, B. Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010. Accessed: 11-05-2019.
- K. Kristensen, A. Nielsen, C. W. Berg, H. Skaug, and B. M. Bell. TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, 70:1–21, 2016.
- P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall, London, UK, 1989.
- A. Zammit-Mangion and N. Cressie. FRK: an R package for spatial and spatio-temporal prediction with large datasets. *Journal of Statistical Software*, In press, 2021.

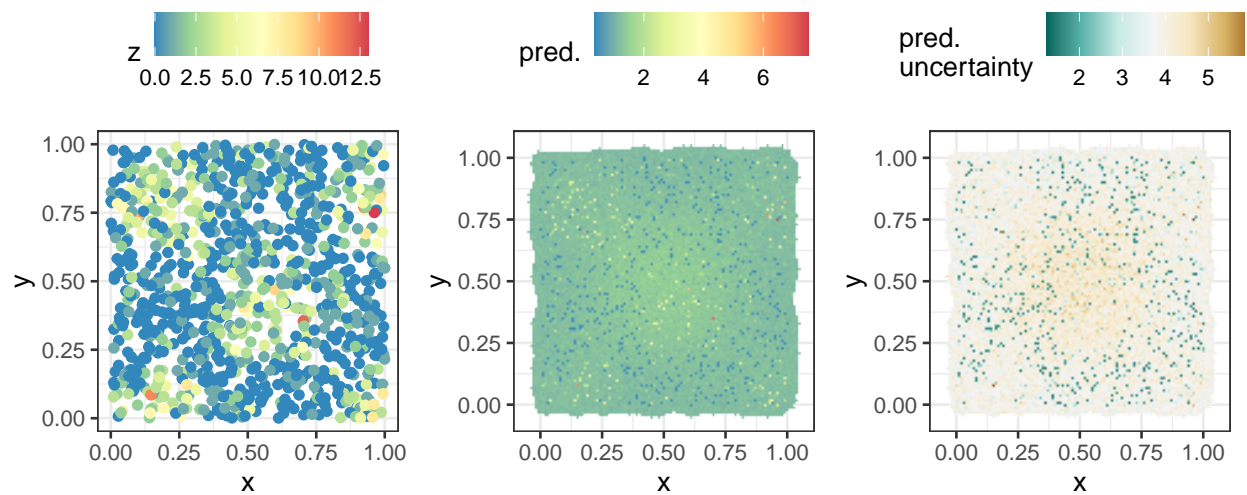


Figure 1: (Left) Simulated poisson data. (Centre) Prediction of the mean process. (Right) Uncertainty quantification of predictions, specifically the width of the 90% posterior predictive interval.