Journal of Statistical Software

April 2008, Volume 25, Issue 10. *http://www.jstatsoft.org/*

Unified Geostatistical Modeling for Data Fusion and Spatial Heteroskedasticity with R Package ramps

Brian J. Smith
The University of Iowa

Jun Yan
University of Connecticut

Mary Kathryn Cowles
The University of Iowa

Abstract

This article illustrates usage of the **ramps** R package, which implements the reparameterized and marginalized posterior sampling (RAMPS) algorithm for complex Bayesian geostatistical models. The RAMPS methodology allows joint modeling of areal and point-source data arising from the same underlying spatial process. A reparametrization of variance parameters facilitates slice sampling based on simplexes, which can be useful in general when multiple variances are present. Prediction at arbitrary points can be made, which is critical in applications where maps are needed. Our implementation takes advantage of sparse matrix operations in the **Matrix** package and can provide substantial savings in computing time for large datasets. A user-friendly interface, similar to the **nlme** mixed effects models package, enables users to analyze datasets with little programming effort. Support is provided for numerous spatial and spatiotemporal correlation structures, user-defined correlation structures, and non-spatial random effects. The package features are illustrated via a synthetic dataset of spatially correlated observation distributed across the state of Iowa, USA.

Keywords: data fusion, geostatistics, heteroskedasticity, RAMPS.

1. Introduction

Spatial data, either areal or geostatistical (point-referenced), are becoming increasingly utilized in the study of many scientific fields due to the accessibility of data monitoring systems and associated datasets. When both types of data are available from the same underlying spatial process, computationally efficient and statistically sound methods are needed for their joint analysis. Markov chain Monte Carlo (MCMC) is a very powerful tool often used for the Bayesian analysis of such data. However, MCMC efficiency can be diminished by substantial autocorrelation in values of the model parameters sampled from the posterior distribution. Yan, Cowles, Wang, and Armstrong (2007) recently proposed a reparameterized and marginalized posterior sampling (RAMPS) algorithm which leads to lower autocorrelation in MCMC samples for Bayesian spatiotemporal geostatistical modeling. The RAMPS algorithm has been further extended to a unified framework of linear mixed models (Cowles, Yan, and Smith 2007) that allows fusion of data obtained at different resolutions (areal and point-referenced) and spatial heteroskedasticity. The general framework also covers cases where prediction at arbitrary sites and non-spatial random effects are needed. This article describes the implementation of the RAMPS algorithm in the R package **ramps** (Smith, Yan, and Cowles 2008) and illustrates its use with a synthetic dataset.

Existing R packages for geostatistical analysis include **fields** (Fields Development Team 2006), **geoR** (Ribeiro and Diggle 2001), **geoRglm** (Christensen and Ribeiro 2002), **gstat** (Pebesma and Wesseling 1998), **sgeostat** (Majure and Gebhardt 2007), **spatial** (Venables and Ripley 2002), and **spBayes** (Finley, Banerjee, and Carlin 2007). The **fields**, **gstat**, **sgeostat**, and **spatial** packages rely on frequentist kriging for modeling and prediction of geostatistical data. The **geoR** (and associated **geoRglm** package for generalized linear models) and **spBayes** packages offer routines to fit Bayesian geostatistical models. These packages do not accommodate combined analysis of point-source and areal data, which is one of the unique features of the **ramps** package. Furthermore, the **spBayes** package is not tailored to yield MCMC samples with lower auto-correlations, which may be critically important in analyzing large datasets. The **geoR** package attains independent posterior samples at the expense of discretizing the prior and posterior densities with respect to the spatial variance and correlation parameters.

The starting point for our unified geostatistical model is the basic RAMPS algorithm for point-source data only, described first in Yan *et al.* (2007). Consider geostatistical observations in a spatial domain D: $\{Y(s_i) : s_i \in D, i = 1, \dots, n\}$, and let $Y = \{Y(s_1), \dots, Y(s_n)\}^\top$. A Gaussian geostatistical model for Y consists of spatial trend, spatial correlation, and measurement error:

$$\begin{aligned} &Y = X\beta + Z + \epsilon, \\ &Z \sim N\left(0, \sigma_z^2 R(\phi)\right), \quad \epsilon \sim N(0, \sigma_e^2 I), \end{aligned} \tag{1}$$

where β is a $p \times 1$ vector of coefficients for covariate matrix $X = \{X^\top(s_1), \dots, X^\top(s_n)\}^\top$, Z is a $n \times 1$ vector capturing the spatial correlation, and ε is a $n \times 1$ vector of independent and identically distributed measurement errors. The distribution of Z is multivariate normal with mean zero and covariance matrix $\sigma_z^2 R(\phi)$, where $R(\phi)$ is the correlation matrix as a function of parameter vector ϕ.

The RAMPS algorithm of Yan *et al.* (2007) includes two steps — reparameterization and marginalization — before drawing samples from the posterior density. The reparameterization step rewrites the model as

$$Y \sim N\left(X\beta, \sigma^2[(1-\kappa)R(\phi) + \kappa I]\right)$$

where $\sigma^2 = \sigma_z^2 + \sigma_e^2$ and $\kappa = \sigma_e^2/\sigma^2$. Letting $\theta = (\phi, \kappa, \sigma^2, \beta)$, the marginalization step factors the posterior density $p(\theta|Y)$ as

$$p(\theta|Y) = p(\phi, \kappa|Y)p(\sigma^2|\phi, \kappa, Y)p(\beta|\phi, \kappa, \sigma^2, Y).$$

With appropriate prior distributions for elements in θ, the second and third distributions on the right hand side can be shown to be inverse gamma and Gaussian, respectively, which makes sampling from them very easy. Conversely, the first distribution is difficult to sample from, and Yan *et al.* (2007) used slice sampling for this critical step.

Cowles *et al.* (2007) subsequently generalized the basic RAMPS algorithm to accommodate the following data complexities and research needs:

1. Fusion of areal data and point-referenced data in a single model. Such data fusion combines data from different sources and resolutions to make more precise statistical inferences, which oftentimes is in terms of narrower credible sets for parameter estimation and prediction.

2. Non-spatial random effects. An example of such non-spatial random effects is the radon data analysis of Smith and Cowles (2007), where many sites have multiple measurements and a site-specific random effect is needed.

3. Multiple variances for each variation source. In fact, data fusion naturally demands multiple variances in the measurement error process for different data sources. The general model framework not only meets this demands but also allows the underlying spatial process to have different variances at different locations; that is, spatial heteroskedasticity.

4. Prediction at arbitrary sites, measured or unmeasured. The RAMPS algorithm can be carried out with very little change in formulation using the method of structured hierarchical models (Hodges 1998; Sargent, Hodges, and Carlin 2000).

All these capabilities are implemented in the **ramps** package, which is publicly available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=ramps`.

The **ramps** package offers a complete set of tools for the conduct of Bayesian geostatistical analysis of large, complex spatial datasets using the RAMPS algorithm. Its unique features are summarized in Table 1 from the aspects of modeling, computing, correlation structures, and user-interface. Note that the correlation structures summarized in Table 1 are a superset of those provided in **nlme** (Pinheiro and Bates 2000) with additional support in **ramps** for great-circle distance as a distance metric option.

This article is organized as follows. Section 2 presents a unified geostatistical model framework that incorporates the aforementioned generalizations. Section 3 discusses some implementational details of the **ramps** package as well as its user interface. Section 4 illustrates the use of the package through a working example. Section 5 reports a performance evaluation of the package in comparison with packages **spBayes** and **geoR** in the context of fitting a simple geostatistical model. A discussion concludes in Section 6.

2. Unified geostatistical model

Let $Y = (Y_a^\top, Y_p^\top)^\top$ be a concatenated vector of areal observations $Y_a = \{Y_{a,1}, \ldots, Y_{a,n_a}\}^\top$ and point-referenced observations $Y_p = \{Y_{p,1}, \ldots, Y_{p,n_p}\}^\top$, where $n_a + n_p = n$ is the total number of observations. The unified Gaussian geostatistical model is

$$\begin{aligned} & Y = X\beta + W\gamma + KZ + \varepsilon \\ & \gamma \sim N\left(0, \Sigma_\gamma\right), \quad Z \sim N(0, \Sigma_Z), \quad \varepsilon \sim N\left(0, \Sigma_\varepsilon\right), \end{aligned} \tag{2}$$

Modeling
• Joint modeling of data from multiple sources (point-source, areal, or both).
• Non-spatial random effects as in a linear mixed model.
• Multiple variances for each variation source (measurement error, spatial, and random effects).
• Prediction at measured or unmeasured sites.
Computing
• Efficient MCMC sampling with the RAMPS algorithm.
• Sparse matrix operation exploited for large datasets.
Correlation structures
• Parametric spatial and spatiotemporal correlation structures, including Gaussian, exponential, powered exponential, spherical, linear, Matérn, rational quadratic, and sine wave.
• Spatial distance calculated as euclidean, great-circle (haversine formula), maximum, or absolute distance.
User interface
• Easy-to-use model specification.
• Object-oriented interface.
• User-extensible spatial correlation structures.
• Three-dimensional spatial plotting of results.

Table 1: Features of the **ramps** package.

where X, W, and K are design matrices for fixed effects β $(p \times 1)$, non-spatial random effects γ $(q \times 1)$, and spatial random effects Z $(S \times 1)$, respectively. The matrix K is defined by

$$K_{ij} = \begin{cases} \frac{1}{N_i}, & \text{site } j \text{ is one of } N_i \text{ measurement sites contributing to } Y_i, \\ 0, & \text{otherwise.} \end{cases}$$

In the case of a point-referenced observation, one measurement site contributes to Y_i, and thus $N_i = 1$. Conversely, multiple measurement sites contribute to an areal observation Y_i. The model defines such an observation as the average over $N_i > 1$ sites. If the actual numbers or locations of contributing sites are unknown, then a uniform grid of spatial sites may be used as an approximation. Accordingly, the N_i will be roughly proportional to the area of the region over which Y_i is an average. The finer the grid of sites; the closer the proportionality will be. In summary, the model formulation (2) accommodates point-referenced data, areal data, multiple measurements, and non-spatial random effects.

Data fusion is made possible in model (2) through the allowance of both areal and point-referenced data. When both types are included simultaneously, a common underlying spatial process $Z(s)$ is assumed, and the design matrix K maps Z contributions to the observed data. For point-referenced data at site s, the contribution from Z is simply $Z(s)$. For data averaged over an area A, the contributions are from $\{Z(s) : s \in G, s \in A\}$, where G is a grid of sites defined over the region from which the data are collected. In practice, the spatial random effects Z in model (2) contain realizations of the spatial process $Z(s)$ at all unique point-referenced and grid sites. The fineness of the grid G can be tuned, depending on the scientific question and computational resources available. Note that Z can also contain realizations at sites that do not contribute to any observed data but at which prediction is of scientific interest, in which case, the corresponding rows in K will consist of zeros; a formulation for this purpose will be presented at the end of this section.

Heteroskedasticity is accommodated by allowing variances to differ across the non-spatial random effects, spatial measurement sites, and individual measurement types. Suppose that there are L_γ different variances for the non-spatial random effects $\sigma^2_{\gamma,i},\ i = 1,\ldots,L_\gamma$; L_Z spatial variances $\sigma^2_{Z,i},\ i = 1,\ldots,L_Z$; and L_ϵ measurement error variances $\sigma^2_{\epsilon,i},\ i = 1,\ldots,L_\epsilon$. Further, let $r_k,\ k = 1,\ldots,q$, be an integer between 1 and L_γ indicating the corresponding random effects variance for γ_k. Likewise, let $v_j, j = 1,\ldots,S$, indicate the spatial variance for observations from site j, and $m_i,\ i = 1,\ldots,n$ the measurement error variance for observation Y_i. We construct vectors for componentwise variances of γ, Z, and ϵ, respectively, as $V_\gamma = \{\sigma^2_{\gamma,r_1},\ldots,\sigma^2_{\gamma,r_q}\}^\top$, $V_Z = \{\sigma^2_{Z,v_1},\ldots,\sigma^2_{Z,v_S}\}^\top$, and $V_\epsilon = \{\sigma^2_{\epsilon,m_1}/w_1,\ldots,\sigma^2_{\epsilon,m_n}/w_n\}^\top$, where w_i, $i = 1,\ldots,n$, is a weight associated with observation i. In the **ramps** package, users may specify the weighting values or accept the default values of 1 for point-source and N_i for areal observations. Assuming exchangeability of random effects, we have $\Sigma_\gamma = \mathrm{diag}(V_\gamma)$, $\Sigma_Z = \mathrm{diag}(\sqrt{V_Z})R(\phi)\,\mathrm{diag}(\sqrt{V_Z})$, and $\Sigma_\epsilon = \mathrm{diag}(V_\epsilon)$, where $R(\phi)$ is a spatial correlation matrix with parameter vector ϕ. This setup is general and allows modeling for spatiotemporal data.

The variance parameters are reparameterized to facilitate the marginalization of the posterior density in the RAMPS algorithm. Concatenate the vectors of measurement error variances, spatial variances, and random effects variances for a total of $F = L_\gamma + L_Z + L_\epsilon$ variance parameters, $\sigma^2_1,\ldots,\sigma^2_F$. If there is one measurement-error variance, one spatial variance, and no random effects variances, then $\sigma^2_1 \equiv \sigma^2_e$ and $\sigma^2_2 \equiv \sigma^2_z$ as in the special case of Yan *et al.* (2007). Our reparameterization is in terms of $\kappa = \{\kappa_1,\ldots,\kappa_F\}^\top$ and σ^2_{tot}, where

$$\sigma^2_{tot} = \sum_{j=1}^{F}\sigma^2_j, \quad \text{and} \quad \kappa_j = \frac{\sigma^2_j}{\sigma^2_{tot}},\ j = 1,2,\ldots,F.$$

Note that $\kappa_F \equiv 1 - \sum_{j=1}^{F-1}\kappa_j$ and is not a free parameter to be estimated. Let $\kappa_\gamma = V_\gamma/\sigma^2_{tot}$, $\kappa_Z = V_Z/\sigma^2_{tot}$, and $\kappa_\epsilon = V_\epsilon/\sigma^2_{tot}$; and define $\Omega_\gamma = \mathrm{diag}(\kappa_\gamma)$, $\Omega_Z = \mathrm{diag}(\sqrt{\kappa_Z})R(\phi)\,\mathrm{diag}(\sqrt{\kappa_Z})$, and $\Omega_\epsilon = \mathrm{diag}(\kappa_\epsilon)$. Then the likelihood can be specified as

$$Y \sim N\left(X\beta,\ \sigma^2_{tot}\Omega\right) \tag{3}$$

where $\Omega = W\Omega_\gamma W^\top + K\Omega_Z K^\top + \Omega_\epsilon$.

Cowles *et al.* (2007) derived the factors of the posterior density $p(\phi,\kappa|Y)$, $p(\sigma^2_{tot}|\phi,\kappa,Y)$, and $p(\beta|\phi,\kappa,\sigma^2_{tot},Y)$. The prior distributions are inverse gamma on σ^2_j, $j = 1,\ldots,F$, flat on β, and uniform for ϕ with appropriately chosen bounds. A challenge presented in sampling from $p(\phi,\kappa|Y)$ is the constraint that κ has support on the standard $(F-1)$-simplex

$$\triangle^{F-1} = \left\{(t_1,\cdots,t_F) \in \mathbb{R}^F \mid \Sigma_i t_i = 1 \text{ and } t_i \geq 0 \text{ for all } i\right\}.$$

Cowles *et al.* (2007) developed a SIMPLICE algorithm, which performs the shrinking step of slice sampling (Neal 2003) on a simplex. A combination of SIMPLICE for κ and Neal's shrinking hyperrectangle slice algorithm for ϕ is implemented in the **ramps** package; see Cowles *et al.* (2007) for details.

The RAMPS procedure can be modified to accommodate prediction at arbitrary sites. Partition Z as $(Z_p^\top, Z_u^\top)^\top$, where Z_p is a vector spatial random effects at sites for which prediction is desired, and Z_u is a vector of spatial random effects at sites for which no prediction is desired. Sampling of β and Z_p can be done in a batch by partitioning and rearranging the matrix K such that $KZ = (K_p, K_u)(Z_p^\top, Z_u^\top)^\top$. Similar to Sargent *et al.* (2000), a stacked linear model can be obtained as

$$\begin{pmatrix} Y \\ 0 \end{pmatrix} = \begin{pmatrix} X & K_p \\ 0 & -I \end{pmatrix} \begin{pmatrix} \beta \\ Z_p \end{pmatrix} + \begin{pmatrix} W\gamma + K_u Z_u + \varepsilon \\ \epsilon_{z_p} \end{pmatrix} \tag{4}$$

or

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}, \tag{5}$$

where $\epsilon_{z_p} \sim N(0, \Sigma_{Z;p,p})$ and $\mathbf{E} \sim N(0, \sigma^2_{tot}\mathbf{\Omega})$ with

$$\mathbf{\Omega} = \begin{pmatrix} W\Omega_\gamma W^\top + K_u\Omega_{Z;u,u}K_u^\top + \Omega_\epsilon & K_u\Omega_{Z;u,p} \\ \Omega_{Z;p,u}K_u^\top & \Omega_{Z;p,p} \end{pmatrix}. \tag{6}$$

This extension simply revises the likelihood expression in equation (3) as

$$\mathbf{Y} \sim N(\mathbf{XB}, \sigma^2_{tot}\mathbf{\Omega}), \tag{7}$$

and the RAMPS algorithm can be applied to sample $\mathbf{B} = (\beta^\top, Z_p^\top)^\top$.

3. Some implementation details

3.1. Correlation structures

Characteristic of geostatistical models is the specification of correlation as a function of the distance between, and possibly orientation of, geographic points in the spatial domain. Our model as implemented in the **ramps** package allows spatial correlation of the general form

$$\left(R\left(\phi\right)\right)_{i,i'} = c\left(s_i, s_{i'}; \phi\right),$$

where $c\left(s_i, s_{i'}; \phi\right)$ is a function of the distance between sites s_i and $s_{i'}$ and the parameter vector ϕ. We provide metrics for the calculation of spatial distance as great-circle distance, Euclidean distance, maximum distance, and sum of absolute differences. Available parametric correlation functions are summarized in Table 3.1. Usage is consistent across the correlation functions, and spatial covariates, such as longitude and latitude, are allowed in the formula specification; see Section 4 for illustration. These are extensions of the **nlme** spatial correlation structures and offer users a consistent interface for geostatistical model specification in the **ramps** package. The spatial correlation structures in **nlme** are not directly used because they do not allow great circle distance, which is very commonly needed for spatial data.

In addition to the supplied functions, users can create their own correlation structures for use with the package by defining a new `corSpatial` class and accompanying `constructor`, `corMatrix`, and `coef` method functions. Examples can be found in the **ramps** source code.

3.2. Model fitting interface

The main user-level function for geostatistical model fitting in the **ramps** package is `georamps` whose definition is given below.

	Spatial correlation		
`corRExp`	Exponential	`corRMatern`	Matérn
`corRExpwr`	Powered exponential	`corRRatio`	Rational quadratic
`corRGaus`	Gaussian	`corRSpher`	Spherical
`corRGneit`	Gneiting	`corRWave`	Sine wave
`corRLin`	Linear		
	Spatiotemporal correlation		
`corRExp2`	Exponential	`corRExpwr2`	Powered exponential
`corRExpwr2Dt`	Temporally-integrated powered exponential		

Table 2: Spatial correlation functions included in the `ramps` package.

```
georamps(fixed, random, correlation, data, subset, weights,
         variance = list(fixed = ~ 1, random = ~ 1, spatial = ~ 1),
         aggregate = list(grid = NULL, blockid = ""),
         control = ramps.control(...), contrasts = NULL, ...)
```

This function implements the RAMPS algorithm for generating samples from the posterior distribution of the model parameters in geostatistical model (2). Model specification of the fixed effects (`fixed`), random effects (`random`), and spatial correlation (`correlation`) arguments mirrors those in package **nlme**. Data fusion and heteroskedasticity are specified by two separate arguments described as follows.

The optional argument `aggregate` allows specification of grid sites over which areal observations are assumed to be aggregated. It is fed by a list of elements: `grid` — a data frame of coordinates to use for Monte Carlo integration over geographic blocks at which areal measurements are available; and `blockid` — a character string specifying the column by which to merge the areal measurements in the data (`data`) with the grid coordinates in `grid`. Merging is performed only for `blockid` values that are common to both datasets. All observations in `data` are treated as point-source measurements by default.

The argument `variance` specifies the types of the multiple variances for each variation source. It is supplied an optional list of one-sided formulas, each of the form `~ g` where `g` defines the grouping factors for: `fixed` — measurement error variances V_ϵ; `random` — random effects error variances V_γ; and `spatial` — spatial variances V_Z. A single variance is assumed in each case by default.

Another important argument is `control`, which controls the fitting process through initial parameter values and prior distributions. Its value must be a `ramps.control` object generated from the `ramps.control` function described next.

3.3. Fitting control

The `ramps.control` function is defined as thus:

```
ramps.control(iter = 1000, beta, sigma2.e, phi, sigma2.z, sigma2.re,
              z.monitor = TRUE, mpdfun = c("mpdbeta", "mpdbetaz"), file)
```

It collects from the user the number of desired MCMC iterations (`iter`), the prior distribution for model parameters (see below), sites at which prediction is needed (`z.monitor`), and file names (`file`) for the output of monitored sample values to external files.

Prior distributions need to be specified with the `param` function for all model parameters — fixed effects `beta`, spatial correlation parameter `phi`, variance parameters for measurement errors `sigma2.e`, spatial random effects `sigma2.z`, and non-spatial random effects `sigma2.re`.

```
param(init, prior = c("flat", "invgamma", "normal", "uniform"),
      tuning = 1, ...)
```

For each group of these parameters, initial values (`init`) and prior density names (`prior`) must be supplied. Prior specifications supported in the current version of the package are as follows: `"flat"` for `beta`, `"invgamma"` for the `sigma2` variances, and `"uniform"` for `phi`. Hyperparameters of the prior distributions are passed through the `'...'` argument.

Tuning of the initial hypercube/simplex sizes for slice sampling is controlled via the `tuning` argument. For spatial correlation parameters ϕ, the tuning parameters can take on positive values and define the factors by which the associated uniform priors are multiplied to define the initial hyperrectangle size. Tuning parameter values to control the initial simplex size for κ may be given in the variance specifications and must be between 0 and 1. The smallest tuning parameter among the variance specifications will be used in tuning the slice algorithm for κ. Smaller values of the tuning parameters will lead to faster sampling at the expense of higher autocorrelation in sampler output.

3.4. Model comparisons

Bayesian model fits are commonly compared with the deviance information criterion (DIC), defined as

$$DIC = E_{\vartheta}\left[D\left(Y \mid \vartheta\right)\right] + p_D$$

where ϑ denotes the model parameters, $D\left(Y \mid \vartheta\right)$ the associated deviance function, and p_D a measure of the effective number of model parameters (Spiegelhalter, Best, Carlin, and van der Linde 2002). Smaller values of the deviance function D are indicative of models that provide better fits to the data. Since this function necessarily decreases as the number of parameters increases, the effective number of parameters p_D is included in the criterion as a penalty term. Consequently, comparisons based on the DIC aim to strike a balance between model goodness-of-fit and parsimony. Preference is given to models with smaller DIC values.

In the context of our geostatistical model, deviance is calculated as a function of the data likelihood

$$Y \sim N(\mathbf{XB}, \sigma^2_{tot}\mathbf{\Omega}).$$

Hence, DIC results are dependent upon the form of the likelihood function used. Two different likelihood formulations are implemented for DIC calculation in the **ramps** package. In the first,

$$\mathbf{X} = X, \quad \mathbf{B} = \beta, \quad \text{and} \quad \mathbf{\Omega} = W\Omega_\gamma W^\top + K\Omega_Z K^\top + \Omega_\epsilon$$

so that the random effects are integrated out of the mean structure, leaving only the fixed covariates and associated β parameters. In the second formulation, both fixed and spatial random effects are included in an SMCMC specification of the likelihood, such that

$$\mathbf{X} = \begin{pmatrix} X & K \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \beta \\ Z \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\Omega} = W\Omega_\gamma W^\top + \Omega_\epsilon.$$

The `mpdfun` argument of the `ramps.control` function determines which one of the two likelihood formulations is used for DIC calculations — values of `"mpdbeta"` and `"mpdbetaz"` correspond to the first and second formulations, respectively. Some guidance on choosing between the two options is provided by the simulation results of Section 4.

3.5. Computational performance

MCMC algorithms for geostatistical models are computationally intensive because the spatial correlation matrix must be updated and decomposed at each sampler iteration. The latter, more expensive operation is of computational order equal to the number of unique spatial sites cubed. Thus, the speed of MCMC algorithms is highly dependent on the performance of the linear algebra routines used. The **ramps** package relies upon core R functions for linear algebra operations. These core function interface with routines provided by the **BLAS** and **LAPACK** linear algebra libraries against which R is linked at runtime. Consequently, one effective way to improve substantially runtime performance of the MCMC algorithm in **ramps** is to link R against optimized versions of **BLAS** and **LAPACK**, such as those provided by ATLAS (Whaley, Petitet, and Dongarra 2001), **ACML** (Advanced Micro Devices, Inc. 2008), Goto **BLAS** (Goto and van de Geijn 2006), or Intel **MKL** (Intel 2008). Details on the configuration of R for these alternative linkages are provided in R's accompanying "Installation and Administration" guide.

Other optimization techniques have been applied to the code directly to improve performance. For example, the K, W, $\mathbf{X}$ and $\boldsymbol{\Omega}$ matrices in the likelihood tend be very sparse, hence suggesting the use of sparse matrix libraries as one way to accelerate computations. Recent versions of the **Matrix** package (Bates and Maechler 2007) provide interfaces to the sparse matrix libraries of Davis (2006) and are used in the implementation of our **ramps** package. As sample size increases, the speed advantages of the **Matrix** routines for sparse matrix operations are well worth the implementation effort. Another example is the choice of marginalized posterior density $p(\phi, \kappa|Y)$ to be used in the SIMPLICE step of the MCMC algorithm. With the `mpdfun` argument of the `ramps.control` function, users can specify that the density be marginalized with respect to the β parameters only (`"mpdbeta"`) or with respect to both the β and Z parameters (`"mpdbetaz"`). The latter choice is recommended as a way to increase algorithm speed when there are multiple observations per measurement site. Regardless of the marginalization choice, samples are drawn from the same posterior distribution.

4. Working example

To illustrate the use of **ramps** for joint analysis of areal and point-source observations, a synthetic dataset was generated from model (2) using the county structure in the state of Iowa, USA. There are $n_a = 99$ counties in the state. Areal observations were generated as county averages from a uniform grid of 391 sites — approximately 4 sites per county. In addition, we generated $n_p = 600$ point-source observations from a set of $n_s = 300$ unique sites sampled from a uniform distribution in Iowa. The multiple point-source measurements per

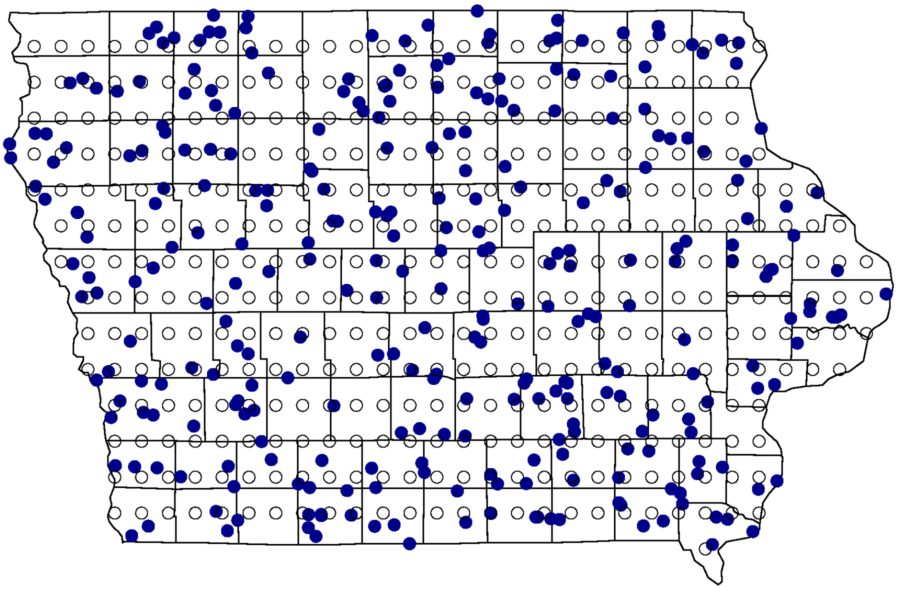


Figure 1: Grid (circles) and point-source (dots) sites included in the synthetic Iowa dataset.

site allow for site-specific non-spatial random effects in the analysis. In Figure 1 the grid of 391 sites is depicted with circles and the 300 point-source measurement sites with dots.

An underlying spatial process was generated from a multivariate normal distribution using an exponential correlation structure with range parameter $\phi = 10$ and variance $\sigma_z^2 = 0.36$. A value of $\phi = 10$ implies that the correlation between two sites drops to 0.05 at a distance of about 30 miles. Measurement errors were generated with variances $\sigma_{\varepsilon,0}^2 = 0.25$ for point-source data and $\sigma_{\varepsilon,1}^2 = 0.09$ for areal data. Site-specific non-spatial random effects were generated with variance $\sigma_\gamma^2 = 0.16$. One fixed effects covariate `areal` is included as an indicator for areal observations. Its β coefficient was set equal to 0.5.

Simulated data are stored in the data frame `simIowa`, with columns `y` for the areal and point-source observations, `areal`, `lon` and `lat` giving the longitude and latitude coordinates, `siteId` as a unique site identifier, and `weights` containing weighting values for measurement error variances. In order to combine the two types of observations in one dataset, `lon`, `lat`, and `siteId` are assigned missing `NA` values for areal observations. A separate grid of measurements sites for areal observations must be supplied to the `georamps` function. The latitude and longitude coordinates of the 391 uniform grid sites in our example are stored in the data frame `simGrid` as variables `lon` and `lat`. An additional indexing variable `id` is included in both `simGrid` and `simIowa` for the purpose of matching grid sites to their respective areal observations.

```
R> print(simIowa)
```

```
    areal           y  id siteId        lon       lat weights
1       1  0.580629645  1      NA        NA        NA       3
2       1  0.780788823  2      NA        NA        NA       3
```

```
3       1  0.568235784   3     NA        NA       NA       5
...
99      1  0.601925872  99     NA        NA       NA       2
100     0  1.291742056 100      1 -93.59640 41.22882       1
101     0  0.247980078 101      1 -93.59640 41.22882       1
102     0 -0.056169094 102      2 -94.93968 41.35889       1
103     0 -0.083346776 103      2 -94.93968 41.35889       1
...
698     0  0.457270781 698    299 -93.80691 42.41017       1
699     0  0.197338314 699    300 -93.50579 42.76213       1
```

```
R> print(simGrid)

          lon      lat id            county
1   -94.64620 41.35573  1        iowa,adair
2   -94.45018 41.35573  1        iowa,adair
3   -94.25416 41.35573  1        iowa,adair
4   -94.84222 40.96397  2        iowa,adams
5   -94.64620 40.96397  2        iowa,adams
6   -94.64620 41.15985  2        iowa,adams
7   -91.50987 43.11865  3    iowa,allamakee
8   -91.31384 43.11865  3    iowa,allamakee
9   -91.50987 43.31453  3    iowa,allamakee
10  -91.31384 43.31453  3    iowa,allamakee
11  -91.11782 43.31453  3    iowa,allamakee
...
390 -93.94283 42.72689 99       iowa,wright
391 -93.75258 42.72689 99       iowa,wright
```

The code below creates a control object of model fitting parameters that must be supplied to the `georamps` function. Prior distributions on θ are: Unif$(1, 60)$ for ϕ, IG$(0.01, 0.01)$ for $\sigma^2_{\epsilon,1}$, $\sigma^2_{\epsilon,2}$, σ^2_z, and σ^2_γ, and flat for β. Also specified are the number of MCMC iterations (`iter`), coordinates of sites at which prediction is desired (`z.monitor`), and optional names of external files to which to save MCMC output for model parameters and spatial random effects (`file`).

```
R> control.fusion <- ramps.control(iter = 1100,
+    phi = param(NA, "uniform", min = 1, max = 60, tuning = 0.5),
+    sigma2.e = param(rep(NA, 2), "invgamma", shape = 0.01, scale = 0.01),
+    sigma2.z = param(NA, "invgamma", shape = 0.01, scale = 0.01),
+    sigma2.re = param(NA, "invgamma", shape = 0.01, scale = 0.01),
+    beta = param(rep(0, 2), "flat"),
+    z.monitor = simGrid[, c("lon", "lat")],
+    mpdfun = "mpdbetaz",
+    file = c("params.txt", "z.txt"))
```

The initial values of all parameters except `beta` are specified as `NA` and, hence, will be generated automatically from the prior distributions. The tuning parameter for ϕ is specified as

0.5, meaning that, in the slice sampling process, the edge width of the initial hyperrectangle for ϕ is one half of the prior range 59.

The joint analysis of areal and point referenced data can now be performed with a call to `georamps`:

```
R> fit.fusion <- georamps(fixed = y ~ areal,
+     random = ~ 1 | siteId,
+     correlation = corRExp(form = ~ lon + lat, metric = "haversine"),
+     variance = list(fixed = ~ areal), weights = weights,
+     data = simIowa,
+     aggregate = list(grid = simGrid, blockid = "id"),
+     control = control.fusion)
```

The model has one covariate (`areal`) as a fixed effect and a site-specific (`siteId`) random effect. The spatial correlation structure is exponential, `corRExp`, with spatial distance computed as great-circle distance (`haversine`). Of note is that, when the `haversine` metric is used, the order of variables must be such that longitude is first and latitude is second. The argument `variance` specifies grouping factors for each variance component associated with the measurement errors (`fixed`), non-spatial random effects (`random`), and spatial random effects (`spatial`). The argument `aggregate` is simply a list which gives the grid from which the areal data are assumed to be obtained and the name of the variable with which the grid and observed data can be merged. The `aggregate` argument would be omitted if analyzing point-source-only data.

We ran 1,100 MCMC iterations and discarded the first 100 as a burn-in sequence. The remaining 1,000 iterations were used for posterior inference. For instance, posterior summaries for the joint analysis were obtained with the code given below.

```
R> fit.fusion1000 <- window(fit.fusion, iter = 101:1100)
R> summary(fit.fusion1000)
```

For comparison, we performed analyses separately for both the point-source only data and the areal only data. The code can be written by modifying that given previously for the fused data analysis and is illustrated in the package help files. Table 4 summarizes the percentiles of the posterior samples from the three analyses. Results in Table 4 indicate that the joint analysis gives narrower 95% credible intervals for parameters common to all analyses; e.g., ϕ and σ^2_Z. Additionally, we used the **ramps** DIC function to evaluate model fits under different specifications of the spatial correlation structure.

```
R> DIC(fit.fusion1000)
```

Results are presented in Table 4 for the two implemented forms of the data likelihood — `mpdbeta` and `mpdbetaz`. Recall that smaller DIC values are suggestive of more desirable models. For selecting among correlation structures, better discrimination is provided by the `mpdbetaz` likelihood. In particular, preference is shown here for the exponential structure which was used to simulate the data. Finally, we excluded the fixed effects indicator for areal observations from the joint model, obtaining DIC values of 1488.5 (`mpdbeta`) and 1921.3 (`mpdbetaz`); both confirming the importance of the indicator in the analysis.

	True	Joint analysis			Point analysis			Areal analysis		
Parameter	values	2.5%	50%	97.5%	2.5%	50%	97.5%	2.5%	50%	97.5%
Intercept	0.00	-0.11	0.03	0.18	-0.12	0.04	0.19			
β	0.50	0.41	0.50	0.60				0.36	0.53	0.68
ϕ	10.00	6.43	10.11	18.58	4.90	11.15	36.81	6.03	12.12	48.34
$\sigma^2_{\epsilon,1}$	0.25	0.23	0.26	0.30	0.22	0.26	0.31			
$\sigma^2_{\epsilon,2}$	0.09	0.01	0.10	0.26				0.01	0.11	0.57
σ^2_z	0.36	0.21	0.33	0.46	0.10	0.28	0.51	0.04	0.30	0.49
σ^2_γ	0.16	0.03	0.17	0.30	0.02	0.21	0.41			

Table 3: Posterior parameter percentiles from joint and separate analyses of the simulated areal and point-source observations.

	Joint analysis		Point analysis		Areal analysis	
Correlation structure	`mpdbeta`	`mpdbetaz`	`mpdbeta`	`mpdbetaz`	`mpdbeta`	`mpdbetaz`
Exponential	1398.4 (6.4)	1604.5 (363.5)	1329.9 (4.3)	1623.9 (279.6)	104.2 (2.4)	0.2 (46.9)
Matérn	1397.5 (5.8)	1693.3 (363.3)	1330.3 (3.7)	1695.0 (270.5)	103.0 (1.1)	26.2 (46.1)
Gaussian	1399.8 (6.6)	1920.8 (393.0)	1329.6 (3.7)	1733.8 (288.6)	103.5 (2.8)	56.4 (58.2)

Table 4: Deviance information criterion values for different spatial correlation specifications, with effect number of parameters given in parentheses. Values are based on likelihood formulations with only fixed effects (`mpdbeta`) and both fixed and spatial random effects (`mpdbetaz`) in the mean structure. The numerical approximation of Gneiting (1999) is used for the Gaussian correlation.

Mapping of the spatial distribution is often of particular interest. There are two ways to get MCMC samples of spatial random effects. The first way is set `z.monitor` in function `ramps.control` equal to `TRUE` or a data frame of coordinates at which prediction is needed. This way is designed for sites that contribute to the observed data. The second way is to use the `predict` method on the `ramps` object returned by `georamps`. This way is designed for sites that do not contributed to the observed data and is particularly useful when prediction on a new grid of sites is needed to draw maps after analyses of point-source data. For example, consider the new grid of sites created with the code given below.

```
R> ia <- map("state", "iowa", plot = FALSE)
R> lon <- seq(min(ia$x, na.rm = TRUE), max(ia$x, na.rm = TRUE), length = 31)
R> lat <- seq(min(ia$y, na.rm = TRUE), max(ia$y, na.rm = TRUE), length = 20)
R> grid <- expand.grid(lon, lat)
```

Prediction of measurements from the point-source process may be obtained with the `fit.fusion1000` objects from the joint analysis via:

```
R> simPred <- data.frame(areal = 0, lon = grid[,1], lat = grid[,2])
```

```
R> pred.fusion0 <- predict(fit.fusion1000, simPred)
```

and prediction of measurements from the areal process via:

```
R> simPred <- data.frame(areal = 1, lon = grid[,1], lat = grid[,2])
R> pred.fusion1 <- predict(fit.fusion1000, simPred)
```

Bayesian output analysis can be carried out to obtain posterior point estimates and credible intervals, which can then be used to produce spatial maps. Figure 2 displays posterior medians and credible interval widths of the predictive distributions from the three analyses, produced with calls to the plotting method in `ramps` of the form:

```
R> plot(pred.fusion0, func = median,
 +    database = "state", regions = "iowa",
 +    resolution = c(155, 100), bw = 0.5,
 +    main = "Joint Analysis, Posterior Median",
 +    xlab = "longitude", ylab = "latitude")

R> plot(pred.fusion0, func = function(x) diff(quantile(x, c(0.025, 0.975))),
 +    database = "state", regions = "iowa",
 +    resolution = c(155, 100), bw = 0.5,
 +    main = "Joint Analysis, 95% Credible Interval Width",
 +    xlab = "longitude", ylab = "latitude")
```

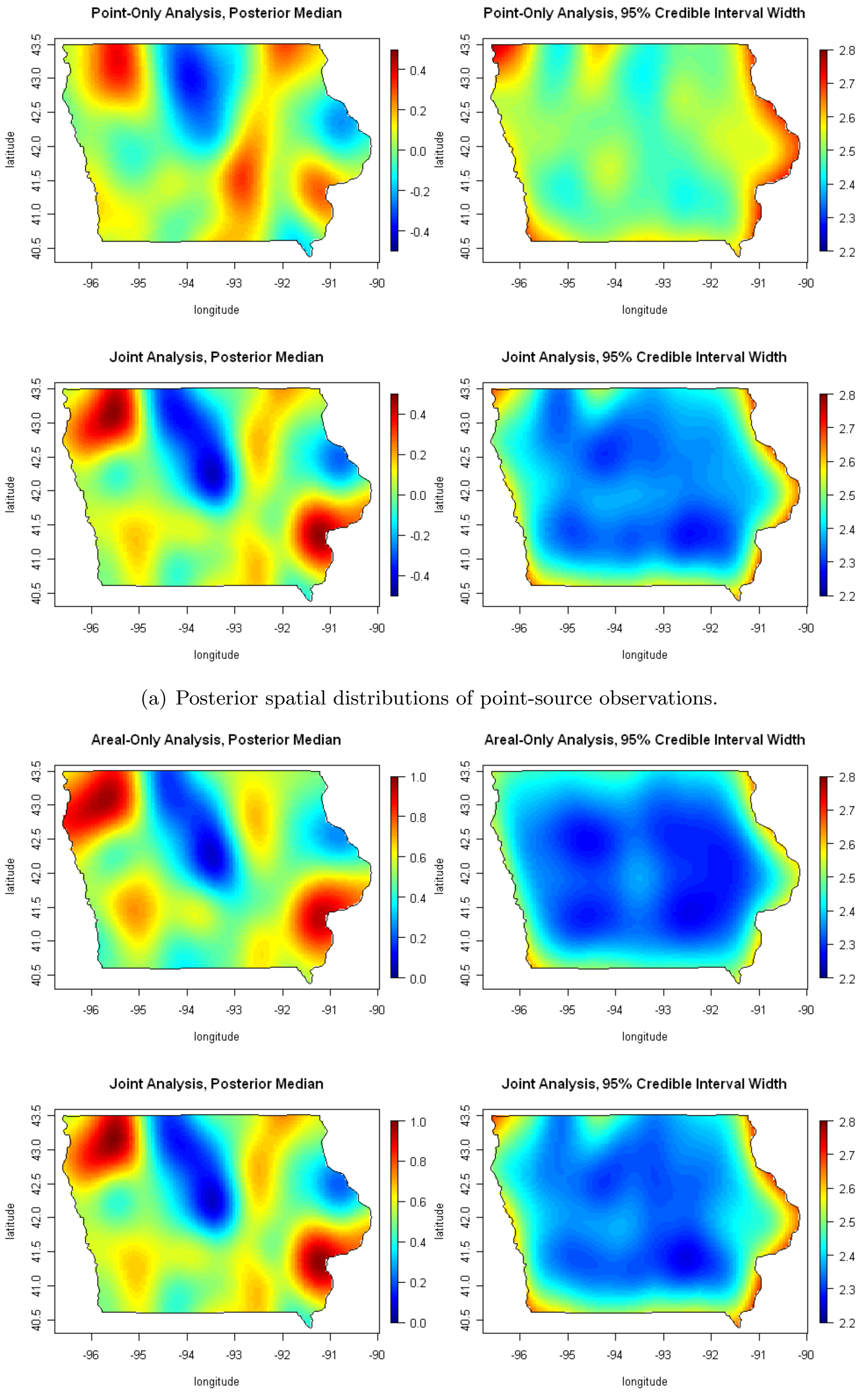
Illustrated in the areal-only data analysis plots is the general tendency of aggregate data to yield overly smooth prediction surfaces. By incorporating the point-source data in the joint analysis, spatial detail is recovered. Furthermore, the combination of both data sources leads to more precise (narrower credible intervals) prediction.

In addition to color image maps of the spatial distribution, the plot function provides a `type` argument that allows for the construction of wireframe (`"w"`) and contour (`"c"`) maps, as shown in the code below and corresponding Figure 3. Note that the plot function easily accommodates other choices of color palettes, such as the one derived here from the HCL color space and described by Zeileis, Hornik, and Murrell (2007).

```
R> col <- rev(heat_hcl(64, l = c(35, 100)))

R> plot(pred.fusion0, type = "w", add.legend = FALSE,
 +    func = median,
 +    database = "state", regions = "iowa",
 +    col = col, resolution = c(45, 30), bw = 0.5, theta = 330, phi = 30,
 +    main = "Joint Analysis, Posterior Median",
 +    xlab = "longitude", ylab = "latitude", zlab = "y")

R> plot(pred.fusion0, type = "c", labcex = 1,
 +    func = median,
 +    database = "state", regions = "iowa",
 +    col = col, resolution = c(155, 100), bw = 0.5,
```


Point-Only Analysis, Posterior Median
Point-Only Analysis, 95% Credible Interval Width
Joint Analysis, Posterior Median
Joint Analysis, 95% Credible Interval Width
Areal-Only Analysis, Posterior Median
Areal-Only Analysis, 95% Credible Interval Width
Joint Analysis, Posterior Median
Joint Analysis, 95% Credible Interval Width
latitude
longitude
40.5
41.0
41.5
42.0
42.5
43.0
43.5
-96
-95
-94
-93
-92
-91
-90
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
2.2
2.3
2.4
2.5
2.6
2.7
2.8

(a) Posterior spatial distributions of point-source observations.

(b) Posterior spatial distributions of areal observations.

Figure 2: Comparison of joint analysis and separate analysis.

```
+    main = "Posterior Median",
+    xlab="longitude", ylab="latitude")

R> plot(pred.fusion0, type = "c", col = col, labcex = 1,
+    func = function(x) diff(quantile(x, c(0.025, 0.975))),
+    database = "state", regions = "iowa",
+    col = col, resolution = c(155, 100), bw = 0.5,
+    main = "95% Credible Interval Width",
+    xlab="longitude", ylab="latitude")
```

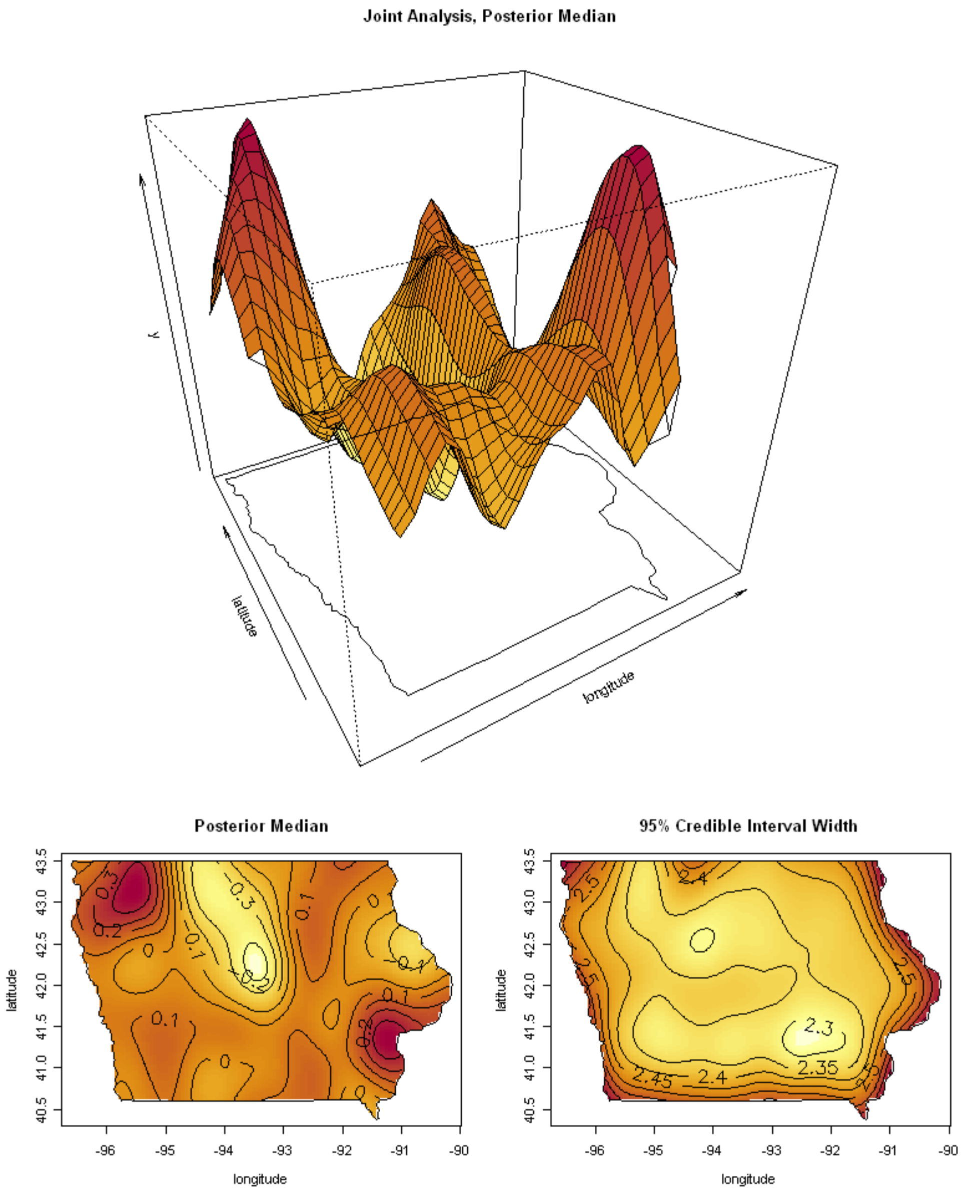


Figure 3: Example wireframe and contour plots available in the **ramps** package.

5. Performance evaluation

Two other R packages, **spBayes** and **geoR**, include functions to perform Bayesian geostatistical analysis to fit the simple model 1, which we use as a platform for performance comparison with **ramps**. In the **spBayes** package, the `ggt.sp` function uses the Metropolis-Hastings-within-Gibbs algorithm to draw samples from the joint posterior density of model parameters. In the **geoR** package, the `krige.bayes` function uses a reparametrized model and a discretization of spatial parameters. Specifically, the likelihood is reparameterized as

$$Y|\beta,\sigma_z^2,\tau_{rel}^2 \sim N\left(X\beta,\sigma_z^2(R(\phi)+\tau_{rel}^2 I)\right) \tag{8}$$

where $\tau_{rel}^2 = \frac{\sigma_e^2}{\sigma_z^2}$. The joint posterior density is marginalized and factored into

$$p(\phi,\tau_{rel}^2|y)p(\sigma_z^2|\phi,\tau_{rel}^2,y)p(\beta|\sigma_z^2,\phi,\tau_{rel}^2,y).$$

Then the domains of ϕ and τ_{rel}^2 are discretized, and the joint posterior marginal density $p(\phi,\tau_{rel}^2|y)$ is evaluated on the resulting two-dimensional grid of values. MCMC sampling then is conducted through the following steps at each iteration, say k: (1) draw the $(\phi^{(k)},\tau_{rel}^{2,(k)})$ pair from the discretized $p(\phi,\tau_{rel}^2|y)$; (2) draw $\sigma_z^{2,(k)}$ from $p(\sigma_z^2|\phi^{(k)},\tau_{rel}^{2,(k)},y)$; (3) draw $\beta^{(k)}$ from $p(\beta|\sigma_z^{2,(k)},\phi^{(k)},\tau_{rel}^{2,(k)},y)$. The result is independent sampling from the discretized joint posterior density.

The `grf` function in **geoR** was used to simulate a dataset of size 800 at sites sampled randomly on $[0,3]\times[0,3]$ with a spherical spatial correlation structure with range parameter $\phi=1$, spatial variance $\sigma_z^2=1$, measurement-error variance $\sigma_e^2=0.5$, and overall mean $\beta=0$.

For the purpose of comparing the performance of the three packages, prior specifications were selected to enable matching of prior densities for the parametrizations in the three packages as closely as possible. Specifically, the prior densities for **ramps** and **spBayes** were chosen as: $\sigma_e^2 \sim \mathrm{IG}(\alpha_e=3,\beta_e=3)$, $\sigma_z^2 \sim \mathrm{IG}(\alpha_z=3,\beta_z=3)$, and $\phi \sim \mathrm{Unif}(1/3,3)$. The correlation parameter ϕ in **spBayes** is the inverse of that in **ramps** and **geoR**, and **spBayes** does not offer the option of a prior on ϕ that is equivalent to the inverse transformation of a uniform distribution. To examine the performance impact of the difference in prior specifications, we also ran **spBayes** using an $\mathrm{IG}(\alpha=4.1333,\beta=2.5067)$ prior on ϕ, where the values of α and β imply a mean and variance equal to that of an inverse $\mathrm{Unif}(1/3,3)$ prior. Using the inverse gamma prior in **spBayes** resulted in a slightly longer run time (55.6 minutes versus 52.8 minutes) and less efficient sampling for the variance parameters than using the uniform prior. The two priors produced similar sampling performance for the β and ϕ parameters. Thus, the difference in priors did not appear to be the main cause of performance differences between between **ramps** and **spBayes**. In the interest of brevity, results are presented only for the uniform prior.

In **geoR**, scaled inverse chi-square is one choice of prior density for σ_z^2; specifying the degrees of freedom as 6 and the variance as 2.25 matched the inverse gamma prior used for σ_z^2 for the other two packages. The parameter ϕ was given a discrete uniform prior on 26 equally-spaced points from 1/3 to 3. Finally, we note that if $\alpha_e=\beta_e$ and $\alpha_z=\beta_z$ in inverse gamma prior densities for σ_e^2 and σ_z^2, then the prior induced on τ_{rel}^2 is $F(2\alpha_z,2\alpha_e)$. Consequently, the prior used for τ_{rel}^2 was discrete on 51 points spanning the 0.005 to 0.99 quantiles of the $F(6,6)$ density, and with probability mass on each point proportional to the $F(6,6)$ density evaluated

	ramps: 73.4 min		**spBayes**: 52.8 min		**geoR**: 124.3 min	
Parameter	ESS	ESS/min	ESS	ESS/min	ESS	ESS/min
β	1000.0	13.62	929.2	17.60	1000	8.06
ϕ	440.5	6.00	45.9	0.87	1000	8.06
σ_e^2	553.9	7.55	37.5	0.71	1000	8.06
σ_z^2	272.8	3.72	36.1	0.68	1000	8.06

Table 5: Performance comparison of effective sample size.

there. Because the posterior density plots for ϕ and σ_e^2 produced from the resulting posterior samples were very jagged due to the coarse grid used for the discretization, a second run was done using 51 prior mass points for ϕ and 101 for τ_{rel}^2. These choices resulted in 51 × 51 combinations of values of ϕ and τ_{rel}^2 at which the joint posterior marginal density of these two parameters had to be evaluated. Results from this finer grid are used for comparison.

MCMC samplers were run for 1,000 iterations using each package, starting from the same initial values of all parameters. All three packages give very similar quantiles of the MCMC samples. The autocorrelations, however, are quite different, which is reflected in terms of "effective sample size" (ESS) (Kass, Carlin, Gelman, and Neal 1998), an established metric for comparing performance of MCMC algorithms. ESS is the number of independent samples that would carry the same amount of information as the available correlated samples. For a given number of MCMC sampler iterations, the higher the autocorrelation in sampler output for a particular parameter, the smaller the resulting effective sample size. Speed of sampling algorithms can be compared fairly in terms of the effective samples per unit run time. The `effectiveSize` function in the R package **coda** (Plummer, Best, Cowles, and Vines 2006) was used to calculate the effective sample size for each parameter from the output of 1,000 MCMC iterations generated with each package.

The ESS and ESS per minute for the 1,000 MCMC samples using the three packages are summarized in Table 5. For the regression coefficient β, all three packages do well, giving 1,000 (or 929 for **spBayes**) independent draws. For the spatial variance σ_z^2, the most difficult parameter to estimate, the **ramps** packages produces a sample worth 272.8 independent draws, about 8 times as many as the **spBayes** package gives (36.1). When time is taken into consideration, the **ramps** package takes 73.4 minutes while the **spBayes** packages takes 52.8 minutes. The **ramps** package gets 5.5 times as many ESS per minute as the **spBayes** package does. The **geoR** package produces higher ESS and ESS/min than the **ramps** package, but recall that the posterior samples are obtained on a grid. The distribution is discrete and the posterior density is jagged.

6. Discussion

The **ramps** package enables Bayesian geostatistical analysis with the very general class of models described by (2). As exemplified in the performance evaluation, its implementation of the RAMPS algorithm provides the advantage of low autocorrelation in MCMC output and therefore more effective samples per unit time than competing methods. The SIMPLICE algorithm which performs slice sampling based on simplexes can be generally useful for cases where multiple variances are present (He, Hodges, and Carlin 2007). As a geostatistical tool, the package also provides smooth maps for either point-source or areal observations. Furthermore, users have full control over specification of the grid from which areal observation are assumed to be drawn.

In our experiments, the spatial correlation parameter ϕ has usually been hardest to estimate and its posterior sample autocorrelation highest among all parameters. Conversely, the fixed effects are easiest to estimate and their posterior samples almost independent. This observation shows the importance of tuning the size of the initial hyperrectangle for ϕ and simplex for κ in the slice sampling by setting `tuning` in `param` when defining the control object with `ramps.control`. For large datasets, one may wish to choose a larger tuning parameter for ϕ and a smaller tuning parameter for κ such that sampling of ϕ traverses the sample space more quickly.

The efficiency of our RAMPS algorithm is determined by the autocorrelation in sampling the marginalized density $p(\phi, \kappa | Y)$. For lower dimensional (ϕ, κ), it is possible to evaluate the density on a grid first, which can then be used to produce close-to-independent samples, as is done in **geoR**. However, for higher dimensional (ϕ, κ), such as arise with spatiotemporal correlation functions and multiple sources of variance, discrete grid evaluation may not be feasible. The RAMPS algorithm and package are designed to accommodate the latter, more general setting. Nevertheless, an adaptive MCMC procedure which takes advantage of past marginalized density evaluations is worth future investigation.

Acknowledgments

The authors thank Vasyl Zhabotynsky for assistance in testing code and evaluating algorithm performance.

References

Advanced Micro Devices, Inc (2008). ***ACML**: AMD Core Math Library.* Advanced Micro Devices, Inc., Numerical Algorithms Group Ltd., Sunnyvale, CA. Version 4.0.0.

Bates D, Maechler M (2007). ***Matrix**: A Matrix Package for R.* R package version 0.999375-7, URL http://CRAN.R-project.org/package=Matrix.

Christensen OF, Ribeiro PJ (2002). "**geoRglm**: A Package for Generalised Linear Spatial Models." *R News*, **2**(2), 26–28. URL http://CRAN.R-project.org/doc/Rnews/.

Cowles MK, Yan J, Smith BJ (2007). "Reparameterized and Marginalized Posterior and Predictive Sampling for Complex Bayesian Geostatistical Models." *Technical Report 384*, The University of Iowa Department of Statistics and Actuarial Science. URL http://www.stat.uiowa.edu/techrep/.

Davis T (2006). *Direct Methods for Sparse Linear Systems.* SIAM.

Fields Development Team (2006). ***fields**: Tools for Spatial Data.* National Center for Atmospheric Research, Boulder, CO. URL http://www.image.ucar.edu/GSP/Software/Fields.

Finley AO, Banerjee S, Carlin BP (2007). "**spBayes**: An R Package for Univariate and Multivariate Hierarchical Point-referenced Spatial Models." *Journal of Statistical Software*, **19**(4), 1–24. URL `http://www.jstatsoft.org/v19/i04/`.

Gneiting T (1999). "Correlation Functions for Atmospheric Data Analysis." *Quarterly Journal of the Royal Meteorological Society*, **125**(559), 2449–2464.

Goto K, van de Geijn R (2006). "High-Performance Implementation of the Level-3 **BLAS**." *Technical Report TR-2006023*, University of Texas at Austin, Department of Computer Science.

He Y, Hodges JS, Carlin BP (2007). "Re-considering the Variance Parameterization in Multiple Precision Models." *Bayesian Analysis*, **2**(3), 529–556.

Hodges JS (1998). "Some Algebra and Geometry for Hierarchical Models, Applied to Diagnostics (Disc: p521-536)." *Journal of the Royal Statistical Society, Series B*, **60**, 497–521.

Intel (2008). *Intel Math Kernel Library Reference Manual*. Intel Corporation, Santa Clara, CA. Version 10.0.

Kass RE, Carlin BP, Gelman A, Neal RM (1998). "Markov Chain Monte Carlo in Practice: A Roundtable Discussion." *The American Statistician*, **52**, 93–100.

Majure JJ, Gebhardt A (2007). ***sgeostat**: An Object-Oriented Framework for Geostatistical Modeling in S+*. R package version 1.0-21, URL `http://CRAN.R-project.org/package=sgeostat`.

Neal RM (2003). "Slice Sampling." *The Annals of Statistics*, **31**(3), 705–767.

Pebesma EJ, Wesseling CG (1998). "**gstat**: A Program for Geostatistical Modelling, Prediction and Simulation." *Computers and Geosciences*, **24**, 17–31.

Pinheiro JC, Bates DM (2000). *Mixed-effects Models in S and S-PLUS*. Springer-Verlag.

Plummer M, Best N, Cowles MK, Vines K (2006). "**CODA**: Convergence Diagnosis and Output Analysis for MCMC." *R News*, **6**(1), 7–11. URL `http://CRAN.R-project.org/doc/Rnews/`.

Ribeiro Paulo J J, Diggle PJ (2001). "**geoR**: A Package for Geostatistical Analysis." *R News*, **1**(2), 14–18. URL `http://CRAN.R-project.org/doc/Rnews/`.

Sargent DJ, Hodges JS, Carlin BP (2000). "Structured Markov Chain Monte Carlo." *Journal of Computational and Graphical Statistics*, **9**(2), 217–234.

Smith BJ, Cowles MK (2007). "Correlating Point-referenced Radon and Areal Uranium Data Arising from a Common Spatial Process." *Journal of the Royal Statistical Society, Series C*, **56**(3), 313–326.

Smith BJ, Yan J, Cowles MK (2008). ***ramps**: Bayesian Geostatistical Modeling with RAMPS*. R package version 0.6-2, URL `http://CRAN.R-project.org/package=ramps`.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002). "Bayesian Measures of Model Complexity and Fit (with Discussion)." *Journal of the Royal Statistical Society, Series B*, **64**(4), 583–639.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer-Verlag.

Whaley RC, Petitet A, Dongarra JJ (2001). "Automated Empirical Optimization of Software and the ATLAS Project." *Parallel Computing*, **27**(1–2), 3–35.

Yan J, Cowles M, Wang S, Armstrong M (2007). "Parallelizing MCMC for Bayesian Spatiotemporal Geostatistical Models." *Statistics and Computing*, **17**(4), 323–335.

Zeileis A, Hornik K, Murrell P (2007). "Escaping RGBland: Selecting Colors for Statistical Graphics." *Report 61*, Department of Statistics and Mathematics, Wirtschaftsuniversität Wien, Research Report Series. URL http://epub.wu-wien.ac.at/dyn/openURL?id=oai:epub.wu-wien.ac.at:epub-wu-01_c87.

Affiliation:

Brian J. Smith
Department of Biostatistics
The University of Iowa
C22 General Hospital
200 Hawkins Drive
Iowa City, IA 52242, United States of America
Email: brian-j-smith@uiowa.edu

Jun Yan
Department of Statistics
University of Connecticut
215 Glenbrook Rd. U-4120
Storrs, CT 06269, United States of America
E-mail: jun.yan@uconn.edu

Mary Kathryn Cowles
Department of Statistics and Actuarial Science
Department of Biostatistics
The University of Iowa
241 Schaeffer Hall
Iowa City, IA 52242, United States of America
E-mail: kcowles@stat.uiowa.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume 25, Issue 10 *Submitted:* 2007-12-17
April 2008 *Accepted:* 2008-03-26