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

Brian J. Smith Jun Yan Mary Kathryn Cowles

November 19, 2007

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 for 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 spatial data. However, its 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, 2007) 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 the associated package **geoRglm** for generalized linear models) and **spBayes** packages offer routines to fit Bayesian geostatistical models. These packages do not accommodate combined analysis of point-source data and areal data, which is one of the unique features of the **ramps** package. 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 of two spatial 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: $\{Y(s_i) : s_i \in D, \quad i = 1, \ldots, n\}$, and let $Y = \{Y(s_1), \ldots, Y(s_n)\}^\top$. A Gaussian geostatistical model for Y consists of spatial trend, spatial correlation, and measurement error:

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

where β is a $p \times 1$ vector of coefficients for covariate matrix $X = \{X^\top(s_1), \ldots, X^\top(s_n)\}^\top$, 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 $\sigma_z^2\Omega(\phi)$, where $\Omega(\phi)$ 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

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

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. The first distribution is very 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) 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. 3) 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. 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.

The **ramps** package offers a comprehensive 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 some of the correlation structures in Table 1 are

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

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

available in **nlme** (Pinheiro and Bates, 2000); but the difference is that the **ramps** package supplies great-circle distance as an option for the distance metric.

This article is organized as follows. Section 2 presents a unified geostatistical model framework that incorporates the aforementioned generalizations; see Cowles *et al.* (2007) for more details about the algorithms. 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 $Y = (Y_a, Y_p)^\top$ 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

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

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 accomodate 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 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 accomodated 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 $\Omega_\gamma = \text{diag}(V_\gamma)$, $\Omega_Z = \text{diag}(V_Z^{1/2})R(\phi)\text{diag}(V_Z^{1/2})$, and $\Omega_\epsilon = \text{diag}(V_\epsilon)$, where $R(\phi)$ is a spatial correlation matrix with parameter vector ϕ. This setup is general and allows modeling for spatio-temporal 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 $\sigma_1^2 \equiv \sigma_e^2$ and $\sigma_2^2 \equiv \sigma_z^2$ 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_{tot}^2, where

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

Note that $\kappa_F \equiv 1 - \sum_{j=1}^{F-1} \kappa_j$ and is not a free parameter to be estimated. Let $\kappa_\gamma = V_\gamma/\sigma_{tot}^2$, $\kappa_Z = V_Z/\sigma_{tot}^2$, and $\kappa_\epsilon = V_\epsilon/\sigma_{tot}^2$. Then the likelihood can be specified as

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

where $\Omega = W\text{diag}(\kappa_\gamma)W^\top + K\text{diag}(\sqrt{\kappa_Z})R(\phi)\text{diag}(\sqrt{\kappa_Z})K^\top + \text{diag}(\kappa_\epsilon)$.

Cowles *et al.* (2007) derived the factors of the posterior density $p(\phi, \kappa|Y)$, $p(\sigma_{tot}^2|\phi, \kappa, Y)$, and $p(\beta|\phi, \kappa, \sigma_{tot}^2, Y)$. The prior distributions are inverse gamma on σ_j^2, $j = 1, \ldots, F$, multivariate normal 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

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

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 $\epsilon_{z_p} \sim N(0, \Omega_{Z;p,p})$, and $\mathbf{E} \sim N(0, \sigma_{tot}^2\mathbf{\Omega})$ 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

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

and the RAMPS algorithm can be applied to sample $\mathbf{B} = (\beta^\top, Z_p^\top)^\top$. The structural formulation (4), in which matrices K, W, $\mathbf{X}$ and $\mathbf{\Omega}$ tend to be be very sparse, suggests 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 advantage of using the **Matrix** package for sparse matrix operations is well worth the implementation effort.

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

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

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 2. 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.

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

	Spatial Correlation		
`corRExp`	exponential	`corRMatern`	Matérn
`corRExpwr`	powered exponential	`corRRatio`	rational quadratic
`corRGaus`	Gaussian	`corRSpher`	spherical
`corRGneit`	Gneiting	`corRWave`	sine wave
`corRLin`	linear		
	Spatio-Temporal Correlation		
`corRExp2`	exponential	`corRExpwr2`	powered exponential
`corRExpwr2Dt`	temporally-integrated powered exponential		

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 source code.

3.2 Model Fitting Inferface

The main user-level function for geostatistical model fitting in the **ramps** package is `georamps`. 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 argument `aggregate` is designed to collect information on what sources of data (areal, point-referenced, or both) are to be analyzed. It is fed by an optional 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 fed by an optional list of one-sided formulas, each of the form `~ g` where `g` defines a grouping factor for the following elements: `fixed` for measurement error variances V_ϵ; `random` for random effects error variances V_γ; and `spatial` for 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 values and prior distributions on the parameters. It is fed by a `ramps.control` object generated from the `ramps.control` function described next.

3.3 Fitting Control

The `ramps.control` function collects from the user the number of desired MCMC iterations (`iter`), the prior distribution for model parameters (see below), optional sites at which prediction is needed (`z.monitor`), and optional file names (`file`) for outputing the monitored sample values.

Prior distributions need to be specified 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`). For each group of these parameters, the `param` function takes inputs of initial values (`init`) and prior density names (`prior`). Four prior distributions are available in the current version: `"flat"`, `"invgamma"`, `"normal"`, or `"uniform"`. Hyperparameters of the prior distributions are passed through the `...` mechanism.

Tuning of the initial sizes of hypercubes/simplexes for slice sampling is specified by the `tuning` argument in the `param` function. This argument takes a value between 0 and 1, which defines the size of the initial hyperrectangle in each MCMC interation for spatial correlation parameters ϕ and the size of the initial simplex for κ. Smaller values of tuning parameters produce faster sampling at the expense of higher autocorrelation in sampler output. Only the first tuning parameter in `sigma2.e` is used for tuning κ. Tuning parameter values are ignored in sampling algorithms for the remaining model parameters.

4 Working Example

To illustrate the use of **ramps** for the 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 Iowa. Areal observations are county averages generated from a uniform grid of 391 sites — approximately 4 sites per county. Point-source observations were generated such that sites may have more than one measurement,

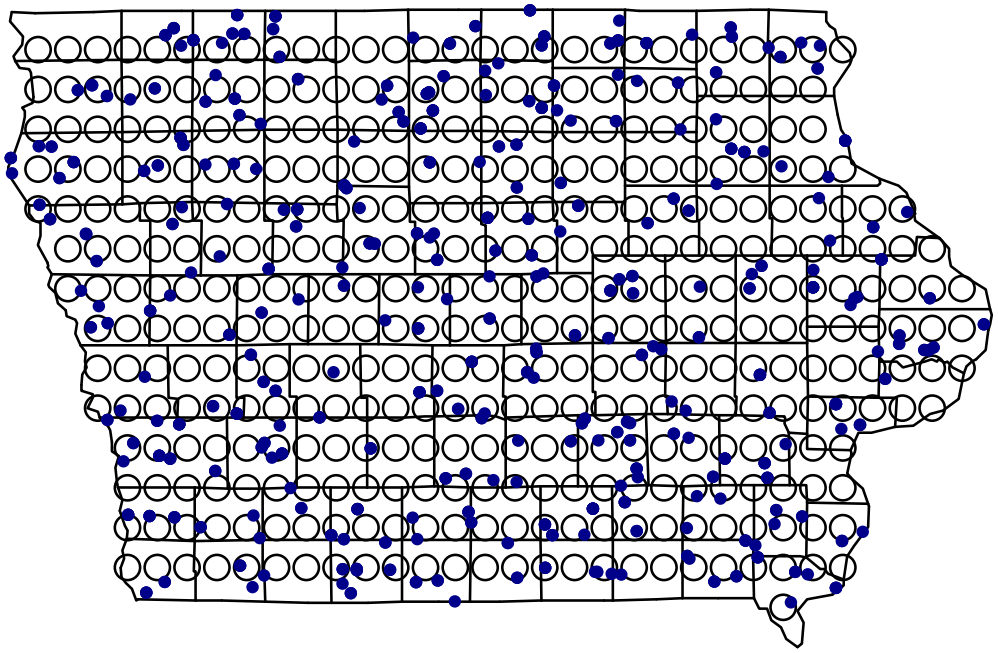


Figure 1: Grid (dots) and point-source (circles) sites included in the synthetic Iowa dataset.

which allows for site-specific non-spatial random effects. There are $n_p = 600$ point-source observations from $n_s = 300$ unique sites. The n_s unique sites were generated from a uniform distribution in Iowa. 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 $\phi = 10$ and variance $\sigma_z^2 = 0.36$. The parameter $\phi = 10$ 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 `weight` 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 -0.056169094 101      2 -94.93968 41.35889       1
102     0 -0.015427496 102      3 -93.26138 42.63638       1
103     0 -0.307796412 103      4 -92.97224 42.65893       1
...
699     0 -1.540805765 699     87 -90.91782 42.44155       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`).

```
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")],
  file = c("params.txt", "z.txt"))
```

The initial values of all parameters except `beta` are specified as `NA` and, hence, will be generated 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`:

```
fit.fusion <- georamps(fixed = y ~ areal,
  random = ~ 1 | siteId,
  correlation = corRExp(form = ~ lon + lat, metric = "haversine"),
  variance = list(fixed = ~ areal),
  data = simIowa, weights = weights,
  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 associted 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 is not used when analyzing point-source-only data.

For comparison, we also performed analyses separately for the point-source data and for the areal data. The code can be written by modifying that given previously for the fused data analysis and is illustrated in the package help files. We ran 1100 MCMC iterations and discarded the first 100. The remaining 1000 iterations were used for posterior inference. For instance, posterior summaries for the joint analysis were obtained with the code given below.

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

Table 3 summarizes the percentiles of the posterior samples from the three analyses. Results in Table 3 indicate that the joint analysis gives narrower 95% credible intervals for parameters common to all analyses; e.g., ϕ and σ_Z^2.

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 grid of sites is needed to draw maps after analyses of point-source data. For example,

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

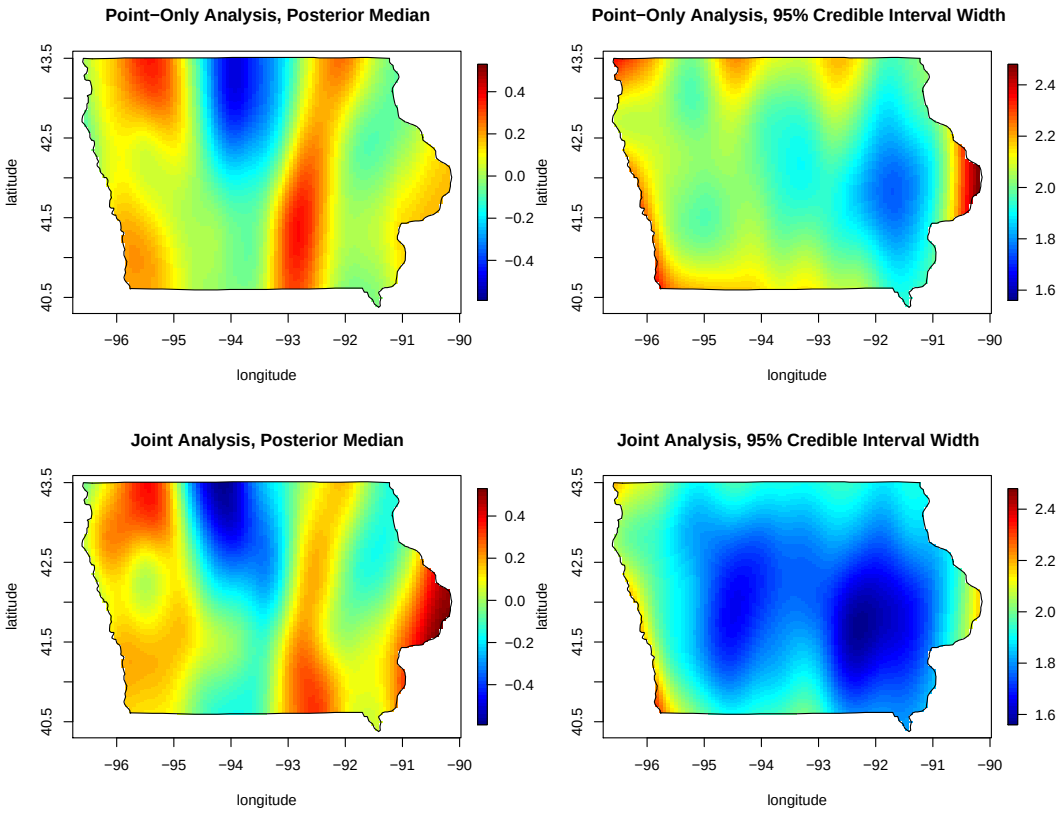
	True	Joint Analysis			Point Data Analysis			Areal Data Analysis		
Parameter	Values	2.5%	50%	97.5%	2.5%	50%	97.5%	2.5%	50%	97.5%
ϕ	10.00	6.57	10.29	20.47	4.50	9.72	31.02	6.65	14.03	54.82
$\sigma^2_{\epsilon,1}$	0.25	0.22	0.26	0.30	0.22	0.26	0.31			
$\sigma^2_{\epsilon,2}$	0.09	0.01	0.09	0.24				0.01	0.17	0.63
σ^2_z	0.36	0.21	0.32	0.47	0.10	0.30	0.53	0.03	0.26	0.47
σ^2_γ	0.16	0.03	0.17	0.30	0.01	0.20	0.40			
Intercept	0.00	−0.11	0.04	0.19	−0.10	0.03	0.18			
β	0.50	0.41	0.50	0.59				0.37	0.53	0.69

`fit.fusion` is the object returned from the joint analysis of areal and point-source data, and thus prediction at a new grid of sites can be obtained via:

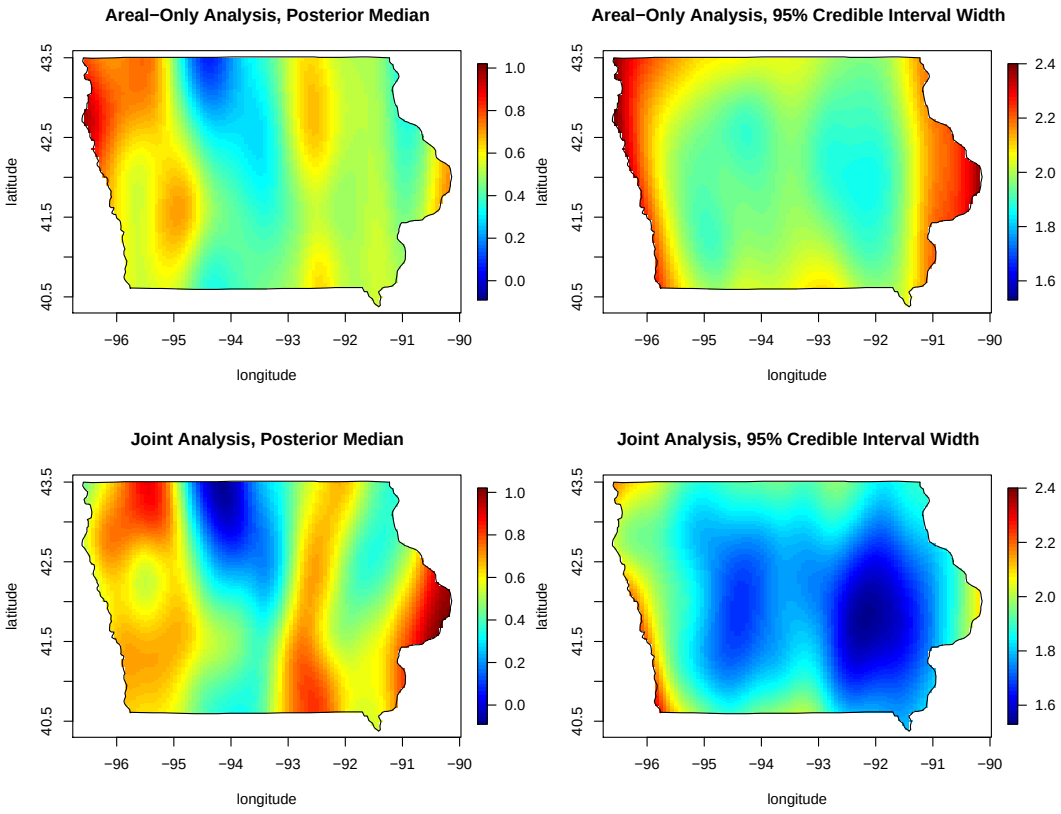
```
## Construct a new grid of spatial sites at which to do prediction
ia <- map("state", "iowa", plot = FALSE)
lon <- seq(min(ia$x, na.rm = TRUE), max(ia$x, na.rm = TRUE), length = 31)
lat <- seq(min(ia$y, na.rm = TRUE), max(ia$y, na.rm = TRUE), length = 20)
grid <- expand.grid(lon, lat)

## Create a data frame for in the predict method function
simPred <- data.frame(lon = grid[,1], lat = grid[,2])

## Obtain prediction for the point-source process
simPred$areal <- 0
pred.fusion0 <- predict(fit.fusion1000, simPred)

## Obtain prediction for the areal process
simPred$areal <- 1
pred.fusion1 <- predict(fit.fusion1000, simPred)
```

Bayesian output analysis (Smith, 2007) 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:

```
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")
```



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



(b) Posterior spatial distributions of areal observations.

Figure 2: Comparison of joint analysis and separate analysis.

```
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 sources of data lead 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.

```
## Wireframe plot of the posterior predictive median
plot(pred.fusion0, type = "w", col = rev(heat.colors(64)), add.legend = F,
     func = function(x) median(x),
     database = "state", regions = "iowa",
     resolution = c(45, 30), bw = 0.5, theta = 330, phi = 30,
     main = "Joint Analysis, Posterior Median",
     xlab = "longitude", ylab = "latitude", zlab = "y")

## Contour plots of the posterior predictive median and interval width
plot(pred.fusion0, type = "c", col = rev(heat.colors(64)), labcex = 1,
     func = function(x) median(x),
     database = "state", regions = "iowa",
     resolution = c(155, 100), bw = 0.5,
     main = "Posterior Median",
     xlab="longitude", ylab="latitude")

plot(pred.areal, type = "c", col = rev(heat.colors(64)), labcex = 1,
     func = function(x) diff(quantile(x, c(0.025, 0.975))),
     database = "state", regions = "iowa",
     resolution = c(155, 100), bw = 0.5,
     main = "95% Credible Interval Width",
     xlab="longitude", ylab="latitude")
```

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 the

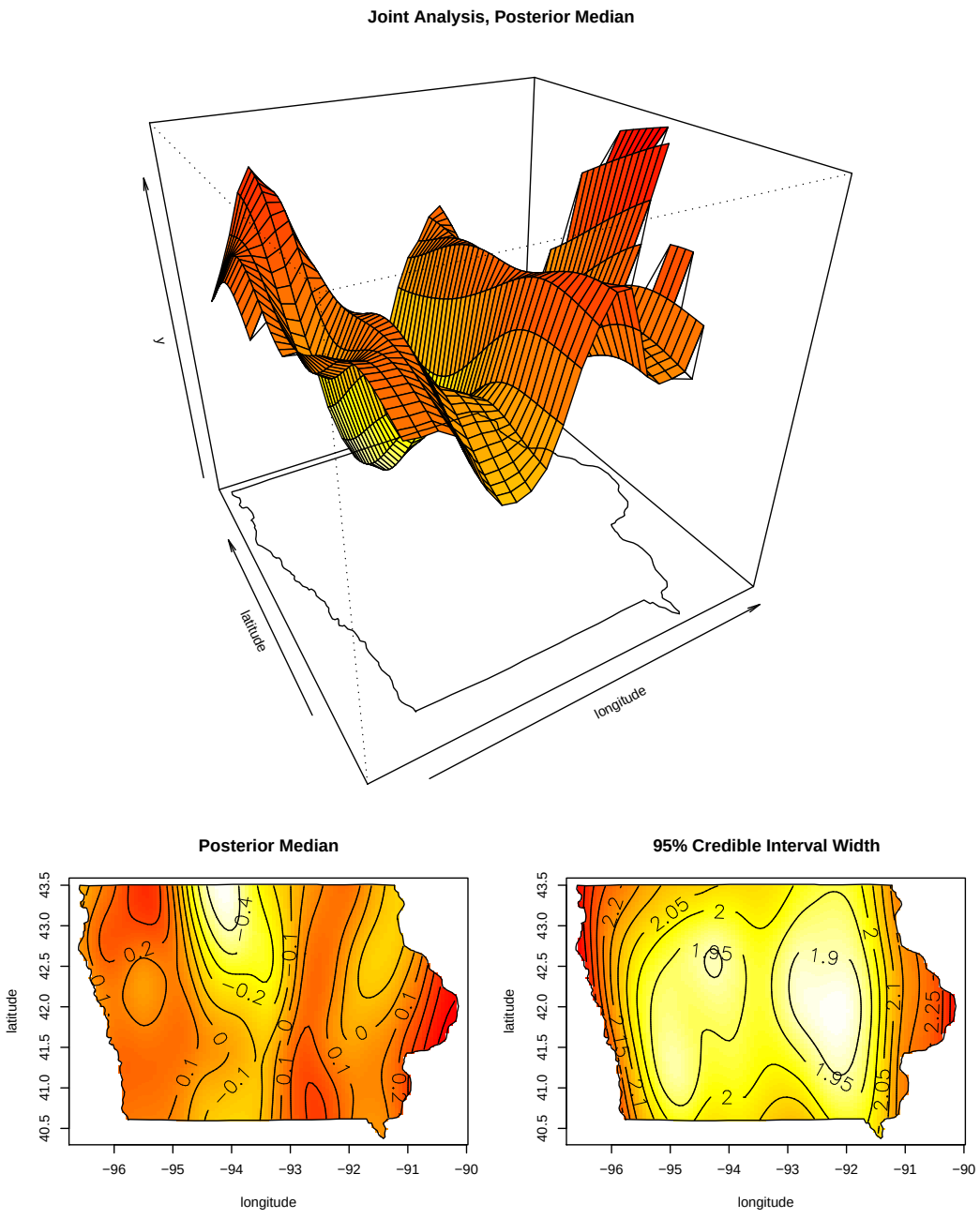


Figure 3: Example wireframe and contour plots available in the **ramps** package.

model parameters. In the **geoR** package, the `krige.bayes` function uses a reparametrized model and two parameters are discretized. Specifically, the likelihood is reparameterized as

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

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 the prior densities for the parameterizations in the three packages as closely as possible. Specifically, the prior densities for **ramps** and **spBayes** were chosen as: $\sigma_e^2 \sim \text{IG}(\alpha_e = 3, \beta_e = 3)$, $\sigma_z^2 \sim \text{IG}(\alpha_z = 3, \beta_z = 3)$, and $\phi \sim \text{Unif}(1/3, 3)$. Because the definition of the correlation parameter ϕ in **spBayes** is the inverse of that in **ramps** and **geoR**, the endpoints of the uniform prior on ϕ were chosen to ameliorate that difference.

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 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 5151 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 1000 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 autocorrelation, 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

Table 4: Comparison results of effective sample size.

	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

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 1000 MCMC iterations generated with each package.

The ESS and ESS per minute for the 1000 MCMC samples using the three packages are summarized in Table 4. For the regression coefficient β, all three packages do well, giving 1000 (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 mant 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 the 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. For higher dimensional (ϕ, κ), however, such grid evaluation may not be feasible. An adaptive procedure which takes advantage of the existing evaluations is worth future investigation.

Acknowledgment

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

References

Bates D, Maechler M (2007). *Matrix: A Matrix package for R*. R package version 0.999375-1.

Christensen OF, Ribeiro PJ (2002). "GeoRglm: A Package for Generalised Linear Spatial Models." *R News*, **2**(2), 26–28.

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, 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.cgd.ucar.edu/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). ISSN 1548-7660. URL `http://www.jstatsoft.org/v19/i04`.

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, Methodological*, **60**, 497–521.

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+*. http://cran.r-project.org/src/contrib/Descriptions/sgeostat.ht ml, 1.0 edition.

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 Inc.

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.

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

Smith BJ (2007). "boa: An R Package for MCMC Output Convergence Assessment and Posterior Inference." *Journal of Statistical Software*, **21**(11). ISSN 1548-7660. URL `http://www.jstatsoft.org/v21/i11`.

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 — Applied Statistics*, **56**(3), 313–326.

Smith BJ, Yan J, Cowles MK (2007). *ramps: Bayesian Geostatistical Modeling with RAMPS*. R package version 0.5-1.

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

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