Journal of Statistical Software

July 2011, Volume 43, Issue 2. *http://www.jstatsoft.org/*

B2Z: An R Package for Bayesian Two-Zone Models

João Vitor Dias Monteiro
University of Minnesota

Sudipto Banerjee
University of Minnesota

Gurumurthy Ramachandran
University of Minnesota

Abstract

A primary issue in industrial hygiene is the estimation of a worker's exposure to chemical, physical and biological agents. Mathematical modeling is increasingly being used as a method for assessing occupational exposures. However, predicting exposure in real settings is constrained by lack of quantitative knowledge of exposure determinants. Recently, Zhang, Banerjee, Yang, Lungu, and Ramachandran (2009) proposed Bayesian hierarchical models for estimating parameters and exposure concentrations for the two-zone differential equation models and for predicting concentrations in a zone near and far away from the source of contamination.

Bayesian estimation, however, can often require substantial amounts of user-defined code and tuning. In this paper, we introduce a statistical software package, **B2Z**, built upon the R statistical computing platform that implements a Bayesian model for estimating model parameters and exposure concentrations in two-zone models. We discuss the algorithms behind our package and illustrate its use with simulated and real data examples.

Keywords: Bayesian inference, two-zone models, Markov chain Monte Carlo, R package.

1. Introduction

The estimation of a worker's exposure to chemical, physical and biological agents is one of the key responsibilities of industrial hygienists (Ramachandran 2005). Statistical and mathematical modeling allows hygienists to systematically evaluate retrospective exposure when past monitoring data are poor or non-existent, to predict current and future exposure in the absence of the working process or operation, and to estimate exposure with only a small number of air samples with possibly high variability. Indeed, Nicas and Jayjock (2002) have argued that modeling may provide more precise estimates of exposure than monitoring with only a few data points. With advances in computational methods and inexpensive software implementation, formal modeling is set to become an indispensable tool in the industrial hygienists' armory.

Zhang *et al.* (2009) recently proposed Bayesian models for estimating model parameters and exposure concentrations for a two-zone model. Their model predicts concentrations in a zone near and far away from the source of contamination. Their model also estimates the contamination rate, air ventilation rate through the system, and the air flow between near and far fields. In their simulation study, they show that the predictions of near field concentration concord with the true values, indicating that the two-zone model assumptions agree with the reality to a large extent and the model is suitable for predicting the contaminant concentration.

It is also well recognized in the statistics literature that spatial *hierarchical* models offer additional richness by building dependencies in different stages. These models follow the Bayesian paradigm of statistical inference (see, e.g., Carlin and Louis 2008; Gelman, Carlin, Stern, and Rubin 2003), where analysis is based upon sampling from the posterior distributions of the different model parameters. Hierarchical models are especially advantageous with data sets having several lurking sources of variation and dependence, where they can estimate much richer models with less stringent assumptions.

In applied research, providing software with a proposed model encourages other researchers to explore the proposed model, detect potential issues and advance methodological research. An exciting prospect in recent times that helps bring such sophisticated statistical methodology to the users is the R project (R Development Core Team 2011). R is a language and environment for statistical computing and graphics that offers several built-in functions for mathematical computations. A convenient feature of R is the ability to create packages (libraries) that implement the new model. In addition, for computationally-intensive tasks, C, C++ and Fortran programs can be linked and invoked by R at run time.

The present paper introduces a R package called **B2Z** – available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=B2Z` – that implements the Bayesian two-zone model proposed by Zhang *et al.* (2009). This package obtains random samples from the posterior distribution of the parameters and exposure concentrations for the Bayesian two-zone model. Currently, three different sampler algorithms are available to do such task: the sampling importance resampling, the incremental mixture importance sampling, and the Metropolis-within-Gibbs sampler. In addition, the package also offers approximate Bayesian estimation using the Bayesian central limit theorem. Section 2 recounts the Bayesian two-zone modeling framework. Section 3 briefly describes the sampler algorithms implemented in **B2Z**. Section 4 illustrates the use of **B2Z** with simulated and real data examples. Finally, Section 5 concludes the paper with some discussion and thoughts.

2. Bayesian two-zone model

Below we describe the Bayesian approach proposed by Zhang *et al.* (2009) for estimating model parameters and exposure concentrations in a two-zone model. The two-zone (also called two-component) model (Nicas 1996; Cherrie and Gelman 1999; Nicas and Miller 1999) assumes the presence of a contamination source in the workplace and that the region very near and around the source is modeled as one well-mixed box, called the *near field*, while the

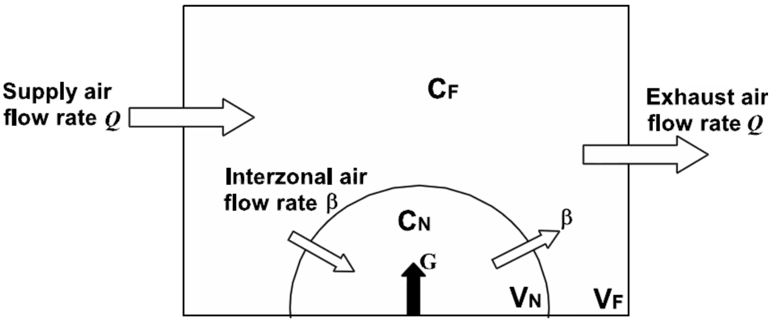

Figure 1: Dynamics of the two-zone model.

rest of the room is another well-mixed box that completely encloses the near field box. This box is called the *far field* and there is some amount of air exchange between the two boxes.

Customarily, it is assumed that each field is a well mixed box, i.e., two distinct places that are in the same field have equal exposure concentration levels. In addition, it is assumed that the contaminant's total mass is emitted at rate G and that there is an airflow rate between the near field and far fields equal to β. The final assumption considers that there are supply and exhaust flow rates which are taken to be the same and equal to Q. Figure 1 is a schematic depiction of the dynamics of the system, where V_N and V_F denote the volumes at the near and far field, respectively. In this context, the occupational hygienist seeks to model the exposure concentrations at the near and far fields based upon observations collected over a period of time. Figure 1, along with the assumptions, yields the following system of differential equations for the two-component model:

$$\frac{d}{dt}\mathbf{C}\left(\boldsymbol{\theta}_1;t\right) = \mathbf{A}\left(\boldsymbol{\theta}_1\right)\mathbf{C}\left(\boldsymbol{\theta}_1;t\right) + \mathbf{g}\left(\boldsymbol{\theta}_1\right) \tag{1}$$

where $\boldsymbol{\theta}_1 = \{\beta, Q, G\}$, $\mathbf{C}\left(\boldsymbol{\theta}_1;t\right) = \begin{bmatrix} C_N\left(\boldsymbol{\theta}_1;t\right) \\ C_F\left(\boldsymbol{\theta}_1;t\right) \end{bmatrix}$, $\mathbf{A}\left(\boldsymbol{\theta}_1\right) = \begin{bmatrix} -\beta/V_N & \beta/V_N \\ \beta/V_F & -(\beta+Q)/V_F \end{bmatrix}$ and $\mathbf{g}\left(\boldsymbol{\theta}_1\right) = \begin{bmatrix} G/V_N \\ 0 \end{bmatrix}$. The functions $C_N\left(\boldsymbol{\theta}_1;t\right)$ and $C_F\left(\boldsymbol{\theta}_1;t\right)$ are the exposure concentrations in the near and far fields at time t, respectively. Equation 1 is the matrix representation of a linear system of ordinary differential equations. When $\mathbf{A}(\boldsymbol{\theta}_1)$ is nonsingular, the solution for (1) has the matrix representation

$$\mathbf{C}(\boldsymbol{\theta}_1;t) = \exp\left(t\mathbf{A}\left(\boldsymbol{\theta}_1\right)\right)\mathbf{C}(\boldsymbol{\theta}_1;0) + \mathbf{A}^{-1}(\boldsymbol{\theta}_1)\left[\exp\left(t\mathbf{A}\left(\boldsymbol{\theta}_1\right)\right) - \mathbf{I}_2\right]\mathbf{g}(\boldsymbol{\theta}_1), \tag{2}$$

where $\mathbf{I}_2$ is the 2×2 identity matrix and $\exp\left(t\mathbf{A}\left(\boldsymbol{\theta}_1\right)\right)$ is the matrix exponential (see, e.g., Laub 2005). Assuming that $C_N\left(\boldsymbol{\theta}_1;0\right) = C_F\left(\boldsymbol{\theta}_1;0\right) = 0$, (2) can be simplified to yield the following unique solution for the far field and near field concentrations:

$$\begin{aligned} C_N\left(\boldsymbol{\theta}_1;t\right) &= \tfrac{G}{Q} + \tfrac{G}{\beta} + G\left(\tfrac{\beta Q+\lambda_2 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_1 t} - G\left(\tfrac{\beta Q+\lambda_1 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_2 t}, \\ C_F\left(\boldsymbol{\theta}_1;t\right) &= \tfrac{G}{Q} + G\left(\tfrac{\lambda_1 V_N+\beta}{\beta}\right)\left(\tfrac{\beta Q+\lambda_2 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_1 t} - G\left(\tfrac{\lambda_2 V_N+\beta}{\beta}\right)\left(\tfrac{\beta Q+\lambda_1 V_N(\beta+Q)}{\beta Q V_N(\lambda_1-\lambda_2)}\right)e^{\lambda_2 t}, \end{aligned} \tag{3}$$

where λ_1 and λ_2 are the eigenvalues of $\mathbf{A}\left(\boldsymbol{\theta}_1\right)$. In fact, these are available in closed form as:

$$\begin{aligned}\lambda_1 &= 0.5\left[-\left(\frac{\beta V_F+(\beta+Q)V_N}{V_N V_F}\right)+\sqrt{\left(\frac{\beta V_F+(\beta+Q)V_N}{V_N V_F}\right)^2-4\left(\frac{\beta Q}{V_N V_F}\right)}\right],\\ \lambda_2 &= 0.5\left[-\left(\frac{\beta V_F+(\beta+Q)V_N}{V_N V_F}\right)-\sqrt{\left(\frac{\beta V_F+(\beta+Q)V_N}{V_N V_F}\right)^2-4\left(\frac{\beta Q}{V_N V_F}\right)}\right].\end{aligned} \tag{4}$$

The solutions for the near and far field zones make intuitive sense. Firstly, notice from (4) that both λ_1 and λ_2 are negative numbers. Thus, the exponential terms in (3) decay asymptotically to zero at large values of t. Consequently, the steady state solution for the far field is G/Q, which is the same as the steady state solution for a one-box model. Also, the steady state solution for the near field is $G/Q+G/\beta$. Therefore, the model predicts relatively higher exposure intensity near the emission source compared to the one-box well mixed room model in steady state conditions. Secondly, if β is less than or equal to Q, then the steady state concentration in the far field is less than twice the steady state concentration in the near field. In general, Q increases relative to β as the room size increases. Thus, the model draws a distinction between exposures of workers near the source and those farther away from the source.

From (3), we also see that the solution of the system in (1) depends upon several parameters. Customarily, V_N and V_F are considered fixed and known, while β, Q and G are regarded as unknown parameters and will need to be estimated. Let $\mathbf{Y}(t)=(Y_N(t),Y_F(t))^\top$ be a 2×1 vector corresponding to the natural logarithm of the exposure concentration at time point t. The observed value of $\mathbf{Y}(t)$ is a combination of two components:

1. *Systematic component*: $\mathbf{C}(\boldsymbol{\theta}_1;t)=(C_N(\boldsymbol{\theta}_1;t),C_F(\boldsymbol{\theta}_1;t))^\top$, the solution of the system of differential equations in (1) at time t;

2. *Measurement error process component*: $\boldsymbol{\epsilon}(t)=(\epsilon_N(t),\epsilon_F(t))^\top$, where $\epsilon_N(t)$ and $\epsilon_F(t)$ are the measurement error processes corresponding to the near and far field, respectively.

This leads to the following measurement model:

$$\mathbf{Y}(t)=\log\mathbf{C}(\boldsymbol{\theta}_1;t)+\boldsymbol{\epsilon}(t), \tag{5}$$

where $\log\mathbf{C}(\boldsymbol{\theta}_1;t)=(\log C_N(\boldsymbol{\theta}_1;t),\log C_F(\boldsymbol{\theta}_1;t))^\top$. Following Zhang *et al.* (2009), we assume Gaussian measurement error and, more specifically, the following two possibilities:

1. *Independent model*: $\boldsymbol{\epsilon}(t)\overset{iid}{\sim}N_2\left(\mathbf{0},\boldsymbol{\Sigma}=\begin{bmatrix}\tau_N & 0\\ 0 & \tau_F\end{bmatrix}\right)$.

2. *Dependent model*: $\boldsymbol{\epsilon}(t)\overset{iid}{\sim}N_2\left(\mathbf{0},\boldsymbol{\Sigma}=\begin{bmatrix}\tau_N & \tau_{NF}\\ \tau_{NF} & \tau_F\end{bmatrix}\right)$.

In the independent model, the measurement errors at the near and far field are assumed to be uncorrelated, while the dependent model relaxes this assumption. For both models, it is assumed that the measurement errors across time are independent and identically distributed.

Let $\mathbf{Y} = \left(\mathbf{Y}(t_1)^\top, \ldots, \mathbf{Y}(t_n)^\top\right)^\top$ denote the $2n \times 1$ vector of observed log-concentrations from the near and far fields at n time points. Letting $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\Sigma}\}$ be the collection of unknown parameters, (5) and the assumptions made on the measurement errors produce the likelihood

$$p(\mathbf{Y} \,|\, \boldsymbol{\theta}) \propto (\det(\boldsymbol{\Sigma}))^{-\frac{n}{2}} \prod_{i=1}^{n} \exp\Big\{ -\frac{1}{2}(\mathbf{Y}(t_i) - \log \mathbf{C}(\boldsymbol{\theta}_1; t_i))^\top \boldsymbol{\Sigma}^{-1} (\mathbf{Y}(t_i) - \log \mathbf{C}(\boldsymbol{\theta}_1; t_i))\Big\}, \tag{6}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the measurement error process. We assume that the components β, Q, G and $\boldsymbol{\Sigma}$ are independent, so that the prior distribution for $\boldsymbol{\theta}$ is $p(\boldsymbol{\theta}) = p(\beta)p(Q)p(G)p(\boldsymbol{\Sigma})$. For the independent model, we assume that $p(\boldsymbol{\Sigma}) = p(\tau_N)p(\tau_F)$ with $\tau_N \sim IG(a_N, b_N)$ and $\tau_F \sim IG(a_F, b_F)$, where a_N and a_F are shape parameters and b_N and b_F are scale parameters for the inverse Gamma distribution. For the dependent model, $\boldsymbol{\Sigma} \sim IW(S, v)$ where S is a scale matrix and v is the degrees of freedom for the inverse Wishart distribution. The parameterizations of the inverse gamma and inverse Wishart here are the same as in Gelman *et al.* (2003). The parameters β, Q and G can have any prior distribution with positive support, i.e., they do not assign positive probabilities to any negative value. Based upon the above assumptions, the posterior distribution of $\boldsymbol{\theta}$ can be computed using Bayes rule as proportional to $p(\boldsymbol{\theta}) \times p(\mathbf{Y} \,|\, \boldsymbol{\theta})$. However, the posterior distribution may not have a closed form precluding analytical inference. Our package **B2Z** has three different sampler algorithms available to obtain samples from the posterior distribution of $\boldsymbol{\theta}$. The algorithms are discussed in the next section.

3. Bayesian estimation

In this section we briefly discuss the three sampling algorithms and the approximation using Bayesian central limit theorem that are available in our package **B2Z**. We also present some algorithmic implementation details.

3.1. Sampling importance resampling

The sampling importance resampling (SIR; Dijk, Hop, and Louter 1987; Rubin 1987, 1988) is a fairly straightforward algorithm used to obtain random samples from a probability distribution, here the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{Y})$. Several variants of this algorithm exist (see, e.g., Robert and Casella 2004), but the basic idea is to sample $\boldsymbol{\theta}$ from an easily tractable distribution (e.g., the prior distribution) so that the SIR tends to choose $\boldsymbol{\theta}_i$'s corresponding to higher values of the likelihood. This sampler is described in the following algorithm:

1. Obtain m i.i.d samples from the prior distribution $p(\boldsymbol{\theta})$. Denote each sample by $\boldsymbol{\theta}_i$, $i = 1, \ldots, m$;

2. For each sample $\boldsymbol{\theta}_i$, evaluate the likelihood $l_i = p(\mathbf{Y} \,|\, \boldsymbol{\theta} = \boldsymbol{\theta}_i)$;

3. Compute the importance weights as:

$$w_i = \frac{l_i}{\sum\limits_{k=1}^{m} l_k};$$

4. From the m samples obtained at the first step, select m samples (with replacement) using the weights w_i's.

In Step (3), the l_i's can be very close to zero so that a large proportion of the importance weights are close to zero as well. To assuage this issue, we implement in our package the following computational trick. We replace the computation of the importance weights in Step (3) for:

$$w_i = \frac{\exp(\widetilde{l}_i)}{\sum\limits_{k=1}^{m} \exp(\widetilde{l}_k)};$$

where $\widetilde{l}_i = \log\left(p(\mathbf{Y} \mid \boldsymbol{\theta} = \boldsymbol{\theta}_i)\right) + C$, and C is a large positive constant. While this may not fully resolve the issue of small weights, it does considerably increase the number of non-zero weights. Nevertheless, for the SIR to sample well from the posterior distribution, m must be large (thousands or even millions) which can be computationally expensive. In fact, in our examples, we often discovered the SIR to be returning very few distinct values, even with an m of size 50,000. This arises due to an inadequate exploration of the parameter domain. Also, it is important to the SIR that the prior distribution agrees with the likelihood. Otherwise very few distinct sampled points from the tails of the prior distribution have sizeable importance weights causing the final sample to have few unique points. The incremental mixture importance sampling, described next, attempts to circumvent these problems.

3.2. Incremental mixture importance sampling

In contrast to the SIR, at each iteration the incremental mixture importance sampling (Steele, Raftery, and Emond 2006; Raftery and Bao 2010, IMIS;) adds samples from a multivariate normal distribution, centered at the point with the highest importance weight, to the current importance sampling distribution. This covers sections of the posterior distribution with high importance weights that are normally underrepresented by the importance sampling distribution. The IMIS algorithm is presented below:

1. Initial stage:
   (a) Draw N_0 i.i.d. samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_0}$ from the prior distribution of $\boldsymbol{\theta}$;

   (b) For each $\boldsymbol{\theta}_i$, evaluate the likelihood $l_i = p(\mathbf{Y} \mid \boldsymbol{\theta} = \boldsymbol{\theta}_i)$ and compute its importance weight as:

   $$w_i^{(0)} = \frac{l_i}{\sum_{k=1}^{N_0} l_k};$$

   (c) $N_1 = N_0$.

2. Importance sampling stage: $k = 1$. **While** some stopping criterion (see below) is not satisfied **do**:
   (a) Denote by $\boldsymbol{\mu}^{(k)}$ the input with highest importance weight among the current importance sample up to iteration k;

   (b) Find the B inputs with smallest Mahalanobis distance to $\boldsymbol{\mu}^{(k)}$. The distances are calculated with respect to the prior covariance matrix of $\boldsymbol{\theta}$, denoted by $\mathbf{V}_{\boldsymbol{\theta}}$. More precisely, the Mahalanobis distance of an input $\mathbf{x}$ to $\boldsymbol{\mu}^{(k)}$ with respect $\mathbf{V}_{\boldsymbol{\theta}}$ is given by:

$$D = \sqrt{\left(\mathbf{x} - \boldsymbol{\mu}^{(k)}\right)^{\top} \mathbf{V}_{\boldsymbol{\theta}}^{-1} \left(\mathbf{x} - \boldsymbol{\mu}^{(k)}\right)}$$

(c) Denote by $u_1, \ldots, u_B$ the importance weights of the B inputs selected in the previous step;

(d) Denote by $\tilde{\boldsymbol{\Sigma}}^{(k)}$ the estimated weighted covariance matrix using the selected B inputs. The weight of the input j is given by:

$$v_j = \frac{(u_j + 1/N_k)}{\sum_{k=1}^{B}(u_j + 1/N_k)} \ \forall \ j = \{1, \ldots, B\};$$

(e) Draw B samples from a $N_d\left(\boldsymbol{\mu}^{(k)}, \tilde{\boldsymbol{\Sigma}}^{(k)}\right)$, where d is the dimension of $\boldsymbol{\theta}$;

(f) Compute the likelihood for each new input from the previous step, and combine the new inputs with the previous ones;

(g) Update: $N_k = N_0 + Bk$;

(h) Compute the mixture sampling distribution $q^{(k)}$ at iteration k, given by:

$$q^{(k)}(\boldsymbol{\theta}_i) = \frac{N_0}{N_k} p(\boldsymbol{\theta}_i) + \frac{B}{N_k} \sum_{s=1}^{k} N_d\left(\boldsymbol{\theta}_i \,|\, \boldsymbol{\mu}^{(s)}, \tilde{\boldsymbol{\Sigma}}^{(s)}\right), \forall \ i = \{1, 2, \ldots, N_k\}$$

where $p(\cdot)$ is the prior distribution of $\boldsymbol{\theta}$ and $N_d(\cdot \,|\, \mathbf{m}, \mathbf{S})$ denotes the multivariate normal density with vector mean $\mathbf{m}$ and covariance matrix $\mathbf{S}$;

(i) Calculate the importance weights using the following formula:

$$w_i^{(k)} = c \times l_i \times \frac{p(\boldsymbol{\theta}_i)}{q^{(k)}(\boldsymbol{\theta}_i)} \ \forall \ i = \{1, 2, \ldots, N_k\}$$

where c is chosen so that the weights sum to 1;

(j) $k = k + 1$.

3. Resample stage: Once the stopping criterion (see below) at the importance sampling stage is satisfied, use the importance weights $w_1^{(K)}, \ldots, w_{N_K}^{(K)}$ to draw, with replacement, M inputs from the importance sample $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_K}$, where K is the total number of iterations at the importance sampling stage.

Stopping criterion: Raftery and Bao (2010) suggest ending the importance sampling step when the expected fraction of unique points in the resample is at least 0.632. **B2Z** follows this suggestion. However, the user can provide a maximum number of iterations at the importance sampling stage in case the stopping criterion takes too long to be met. Raftery and Bao (2010) also suggest that a good choice for the input parameters is: $N_0 = 1000d$, $B = 100d$ and $M = 3000$. Recall that if the independent model is considered $d = 5$, otherwise $d = 6$.

3.3. Metropolis-within-Gibbs sampling

Gibbs sampling (Geman and Geman 1984; Gelfand and Smith 1990) is a popular Markov chain Monte Carlo (MCMC) algorithm that samples from the full conditional distributions for each parameter. This is attractive in our context since the full conditional distributions for τ_N and τ_F in the independent model and for $\mathbf{\Sigma}$ in the dependent model are respectively given by:

1. *Independent model*:

$$\tau_N \,|\, \boldsymbol{\theta}_1, \mathbf{Y} \sim IG\left(a_N + \frac{n}{2},\, b_N + \frac{1}{2}\sum_{j=1}^{n}(Y_N(t_j) - \log(C_N(\boldsymbol{\theta}_1; t_j)))^2\right),$$
$$\tau_F \,|\, \boldsymbol{\theta}_1, \mathbf{Y} \sim IG\left(a_F + \frac{n}{2},\, b_F + \frac{1}{2}\sum_{j=1}^{n}(Y_F(t_j) - \log(C_F(\boldsymbol{\theta}_1; t_j)))^2\right).$$

2. *Dependent model*: $\mathbf{\Sigma} \,|\, \boldsymbol{\theta}_1, \mathbf{Y} \sim IW(S_1, \nu_1)$, where
   (a) $S_1 = S + \sum_{j=1}^{n} (\mathbf{Y}(t_j) - \log(\mathbf{C}(\boldsymbol{\theta}_1; t_j)))\,(\mathbf{Y}(t_j) - \log(\mathbf{C}(\boldsymbol{\theta}_1; t_j)))^\top$;
   (b) $\nu_1 = \nu + n$.

However, the full conditional distribution of $\boldsymbol{\theta}_1$ does not have a closed form and we sample from its full conditional distribution using the Metropolis algorithm (see Metropolis algorithm section below). This is called the Gibbs sampler with Metropolis step (or Metropolis-within-Gibbs). The algorithm is as follows:

```
Provide the initial value $\boldsymbol{\theta}_1^{(0)}$;
for $k$ in $1:N$ do
  Draw a sample from $\mathbf{\Sigma} \,|\, \boldsymbol{\theta}_1^{(k-1)}, \mathbf{Y}$, and denote it as $\mathbf{\Sigma}^{(k)}$
  Using Metropolis sampler, draw a sample from $\boldsymbol{\theta}_1 \,|\, \mathbf{\Sigma}^{(k)}, \mathbf{Y}$ and denote by $\boldsymbol{\theta}^{(k)}$.
end for
```

Metropolis algorithm

Metropolis (Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953; Hastings 1970) is a well known MCMC sampling algorithm. Here, at each iteration we sample a candidate from a proposal distribution and then decide whether the candidate should be accepted or not. This decision is based on the ratio of the posterior distribution evaluated at the candidate and the previously accepted candidate. Since this is a ratio one needs to evaluate the posterior distribution only up to a proportionality constant. Several variants of the Metropolis sampler exist (see, e.g., Robert and Casella 2004). The one that is currently implemented in **B2Z** is the *random-walk* Metropolis algorithm with normal proposals and is described as follows:

```
Provide the initial value $\boldsymbol{\theta}_1^{(0)}$;
for $k$ in $1:N$ do
  Generate a candidate $\boldsymbol{\theta}_1^{(*)}$ from a $N_d\left(\boldsymbol{\theta}_1^{(k-1)}, \mathbf{V}\right)$
  $r = \frac{p\left(\boldsymbol{\theta}_1^{(*)} \middle| \mathbf{Y}\right)}{p\left(\boldsymbol{\theta}_1^{(k-1)} \middle| \mathbf{Y}\right)}$
  if $r \geq 1$ then
    $\boldsymbol{\theta}_1^{(k)} \leftarrow \boldsymbol{\theta}_1^{(*)}$
  else
```

```
      Generate a number $u$ from a $U(0,1)$
      if $u < r$ then
        $\boldsymbol{\theta}_1^{(k)} \leftarrow \boldsymbol{\theta}_1^{(*)}$
      else
        $\boldsymbol{\theta}_1^{(k)} \leftarrow \boldsymbol{\theta}_1^{(k-1)}$
      end if
    end if
  end for
```

The input parameters for the Gibbs sampler with the Metropolis step algorithm are the number of updates N, the vector initial value $\boldsymbol{\theta}_1^{(0)}$, and a covariance matrix $\mathbf{V}$ for the proposal distribution. An approach that usually works well in practice estimates the posterior mode and uses it as an initial value. For $\mathbf{V}$, we use the negative inverse of the hessian matrix of the log posterior distribution evaluated at the posterior mode. This approach is implemented in **B2Z** as the default for setting initial values and specifying the proposal covariance matrix $\mathbf{V}$.

3.4. Bayesian central limit theorem

Differently from the previous sections where we discussed about samplers algorithms, in this section we briefly discuss the Bayesian central limit theorem (BCLT). This theorem states that under some assumptions we can use a Gaussian approximation to the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathbf{Y}) = \dfrac{f(\boldsymbol{\theta})}{\int f(\boldsymbol{\theta}) d\boldsymbol{\theta}}$, where $f(\boldsymbol{\theta}) = p(\mathbf{Y} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta})$.

Consider a Taylor expansion of $\ln(f(\boldsymbol{\theta}))$ centered on the posterior mode $\boldsymbol{\theta}_0$. At $\boldsymbol{\theta}_0$ the gradient $\nabla f(\theta)$ will vanish. Thus the expansion around $\boldsymbol{\theta}_0$ is given by

$$\ln(f(\boldsymbol{\theta})) \simeq \ln(f(\boldsymbol{\theta}_0)) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \mathbf{H} (\boldsymbol{\theta} - \boldsymbol{\theta}_0) \ , \tag{7}$$

where $\mathbf{H}$ is the negative Hessian matrix of the log posterior distribution evaluated at the posterior mode. Exponentiating both sides in Equation 7, we obtain

$$f(\boldsymbol{\theta}) \simeq f(\boldsymbol{\theta}_0) \exp\left\{ -\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \mathbf{H} (\boldsymbol{\theta} - \boldsymbol{\theta}_0) \right\} \ . \tag{8}$$

From (8), $f(\boldsymbol{\theta})$ is seen to be approximately equal to a multivariate normal density with mean $\boldsymbol{\theta}_0$ and covariance matrix $\mathbf{H}^{-1}$. Since the posterior density, $p(\boldsymbol{\theta} \,|\, \mathbf{Y})$, is proportional to $f(\boldsymbol{\theta})$, it too is approximately equal to the multivariate normal density. We note this approximation assumes that the prior distribution of $\boldsymbol{\theta}$ and the likelihood must be positive and twice differentiable near the posterior mode. For further details, see Bishop (2006).

To compute estimates of the parameters using the BCLT, we use the R built-in function called `nlminb`. This function implements constrained and unconstrained optimizations using PORT routines (Gay 1990), allowing us to estimate the posterior mode numerically. Subsequently, we use the R function `hessian`, from the package **numDeriv** (Gilbert 2011) to calculate a numerical approximation to the Hessian matrix of the log posterior function at the estimated posterior mode.

3.5. Algorithmic implementation details

B2Z is an R package that performs sampling-based Bayesian inference for the two-zone model described in Section 4.2. Currently, three sampling algorithms are available: (a) MCMC,

(b) IMIS and (c) SIR. In addition, the package also offers approximate Bayesian estimation using the (d) BCLT. The Bayesian two-zone model can be fitted using the function `B2ZM`, where the desired sampling algorithm is specified as an argument to this function. Another option is to use one of the following functions directly: `B2ZM_BCLT`, `B2ZM_MCMC`, `B2ZM_IMIS` and `B2ZM_SIR`. In either one of the cases, the output is a valid input for the functions `summary` and `plot`. For instance, suppose `fit` is an output from `B2ZM`. Then, the line command `summary(fit)` returns the following:

- Some posterior summaries for each of the parameters $\boldsymbol{\theta}$:
  - Posterior median, mean, standard deviation;
  - $100(1-\alpha)\%$ credibile intervals, where α is specified by the user;
  - Posterior covariance matrix.
- Posterior model comparisons using the deviance information criterion (DIC); see Spiegelhalter, Best, Carlin, and van der Linde (2002).
- Sample quality measurements that depend on the sampler algorithm. Specifically,
  - SIR: Effective sample size (ESS), proportion of unique points in the sample, maximum importance weight;
  - IMIS: ESS, maximum importance weight, variance of the re-scaled importance weights, entropy of importance weights relative to uniformity, expected fraction of unique points and expected number of unique points after re-sampling;
  - MCMC: effective sample size and MCMC acceptance rate.

The package **coda** (Plummer, Best, Cowles, and Vines 2006) offers several other diagnostics measures. We show in Section 4.1 how to integrate the packages **B2Z** and **coda**. For details on some of the above quantities (e.g., DIC and ESS) see Carlin and Louis (2008).

The line command `plot(fit)` produces some graphical summaries of the estimated model. In particular, this line command returns:

- $100(1-\alpha)\%$ posterior predictive interval along with the posterior median of the log concentrations at the near field over the observed period of time, where α is specified by the user;
- $100(1-\alpha)\%$ posterior predictive interval and the posterior median of the log concentrations at the far field over the observed period of time, where α is specified by the user;
- Empirical posterior distributions for each parameter in the model;
- If Metropolis-within-Gibbs is selected, autocorrelation function (ACF) and trace history of the sampling of each parameter is also plotted.

Due to the domain of the parameters in the model, we actually implement the algorithms cited previously (except SIR) on a transformation of $\boldsymbol{\theta}$. After sampling from the posterior distribution of the transformed variables, we back transform to obtain a sample from the posterior distribution of $\boldsymbol{\theta}$. In particular, consider the dependent model and denote $X_1 = \beta$, $X_2 = Q$, $X_3 = G$, $X_4 = \tau_N$, $X_5 = \tau_F$ and $X_6 = \tau_{NF}$. Suppose the priors for β, Q and G have supports (a_1, b_1), (a_2, b_2) and (a_3, b_3), respectively, where $0 \leq a_i < b_i < \infty$ for all $i = 1, \ldots, 3$. Consider the following transformations given by $h_i(\cdot)$ for $i = 1, 2, \ldots, 6$:

$$\begin{aligned}
U_i &= h_i(X_i) = \log\left(\frac{X_i - a_i}{b_i - X_i}\right) \ \ \forall \ \ i = \{1, 2, 3\}, \\
U_i &= h_i(X_i) = \log(X_i) \ \ \forall \ \ i = \{4, 5\}, \\
U_6 &= h_6(X_4, X_5, X_6) = \log\left(\frac{X_6 - \sqrt{X_4 X_5}}{\sqrt{X_4 X_5} - X_6}\right) .
\end{aligned}$$

Therefore, the domain of $\mathbf{U} = (U_1, U_2, \ldots, U_6)^\top$ is in $\mathbb{R}^6$. Notice that if the independent model is considered, the variable U_6 is not needed in $\mathbf{U}$, and therefore its domain is $\mathbb{R}^5$. Thus, the density of $\mathbf{U}$ is given by:

$$p(\mathbf{U} = \mathbf{u} \,|\, \mathbf{Y}) = p(\boldsymbol{\theta} = \mathbf{h}^{-1}(\mathbf{u}) \,|\, \mathbf{Y}) \cdot |\mathbf{J}|, \tag{9}$$

where $\mathbf{h}^{-1}(\mathbf{u}) = (h_1^{-1}(u_1), h_2^{-1}(u_2), \ldots, h_6^{-1}(u_6))^\top$ and $|\mathbf{J}|$ is the jacobian of the transformation with $\mathbf{J}$ being the 6×6 matrix whose (i, j)-th element is $J_{ij} = \frac{\partial X_i}{\partial U_j}$. Regardless of the model choice, the matrix $\mathbf{J}$ is a lower triangular matrix, therefore $|\mathbf{J}|$ is the product of the diagonal elements, which is

$$\begin{aligned}
|\mathbf{J}| &= \prod_{i=1}^{3} \frac{(b_i - a_i)e^{U_i}}{(1 + e^{U_i})^2} \times \prod_{i=4}^{5} e^{U_i}, && \text{(independent model)} \\
|\mathbf{J}| &= \prod_{i=1}^{3} \frac{(b_i - a_i)e^{U_i}}{(1 + e^{U_i})^2} \times \prod_{i=4}^{5} e^{U_i} \times \frac{2e^{\frac{(U_4+U_5)}{2} + U_6}}{(1 + e^{U_6})^2}. && \text{(dependent model)}
\end{aligned} \tag{10}$$

The transformation $h_i(\cdot)$ makes the support of the U_i the real line, which improves the algorithmic efficiency. Also, when we back transform, the sampled values for β, Q and G are within their respective domain, and the covariance matrices composed by the sampled values for τ_N, τ_F and τ_{NF} are positive definite, since h_4 and h_5 guarantee that τ_N and τ_F are positive, and h_6 ensures that the covariance inequality is held, that is, $\tau_{NF}^2 \leq \tau_N \tau_F$.

When any of the domains of β, Q or G is $(0, \infty)$, the corresponding U_i is the natural logarithm of X_i and the computation of the jacobian is still very similar to the one presented in Equation 10.

4. Illustrating B2Z

In this section we illustrate **B2Z** using two synthetic datasets and a real dataset. The simulated exposure concentrations at the near and far fields, over n time points, were generated according to the following algorithm:

1. Choose the values of the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\Sigma}$ as desired. Recall that $\boldsymbol{\Sigma}$ is a diagonal matrix in the independent model, or a matrix with non-null entries in the off diagonal for the dependent model. In any case, $\boldsymbol{\Sigma}$ must be a positive definite matrix.

2. **For** $(i \text{ in } 1:n)$

   (a) Using the fixed parameters in *Step 1*, find the solution of the system of differential equations in (1). Denote this solution by

   $$\mathbf{C}\left(\boldsymbol{\theta}_1; t_i\right) = \left(C_N\left(\boldsymbol{\theta}_1; t_i\right), C_F\left(\boldsymbol{\theta}_1; t_i\right)\right)^\top.$$

   (b) Generate the measurement error component $\boldsymbol{\epsilon}(t_i) = (\epsilon_N(t_i), \epsilon_F(t_i))^\top$ from a $N_2(\mathbf{0}, \boldsymbol{\Sigma})$.

   (c) The log exposure concentrations in the near and far fields at time t_i are

   $$\mathbf{Y}(t_i) = \log \mathbf{C}\left(\boldsymbol{\theta}_1; t_i\right) + \boldsymbol{\epsilon}(t_i).$$

   (d) The exposure concentrations in the near and far fields at time t_i are $\exp(\mathbf{Y}(t_i))$.

The dataset in Section 4.1 was generated considering dependent measurement errors, i.e., $\tau_{NF} \neq 0$. Section 4.2 presents an application of the Bayesian two-zone model to a simulated dataset where the measurement errors are independent, while Section 4.3 applies the model to a real exposure dataset. We started each sampler with a seed set to 2011.

4.1. Simulated data 1

Consider a simulated dataset that contains 100 exposure concentrations equally-spaced between times 0 and 4 minutes. Following the study simulation in Zhang *et al.* (2009), the parameters values used in the simulation process are: $\beta = 5\ m^3/min$, $Q = 13.8\ m^3/min$, $G = 351.54\ mg/min$ and $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 1 & 0.5 \\ 0.5 & 0.64 \end{smallmatrix}\right]$. The volumes at the near and far fields in this simulated data are, respectively, $V_N = \pi \times 10^{-3} m^3$ and $V_F = 3.8 m^3$.

To fit the Bayesian two-zone model, we need to specify the prior distributions for the unknown parameters. We assume that $\beta \sim \mathit{Unif}(0, 10)$, $Q \sim \mathit{Unif}(11, 17)$ and $G \sim \mathit{Unif}(281, 482)$. The dependent model is used in this section. Therefore, we assume that $\boldsymbol{\Sigma} \sim IW\left(\left[\begin{smallmatrix} 10 & 0 \\ 0 & 10 \end{smallmatrix}\right], 4\right)$.

The example code below illustrates how to specify the model information and the sampling algorithm desired using **B2Z**.

```
R> set.seed(2011)
R> fit.depend <- B2ZM(data = ex1, priorBeta = "unif(0,10)",
+    indep.model = FALSE, priorQ = "unif(11,17)", priorG = "unif(281,482)",
+    S = diag(10,2), v = 4, VN = pi * 10^-3, VF = 3.8, sampler = "MCMC",
+    mcmc.control = list(NUpd = 10000, burnin = 1000, lag = 1, m = 5000))
```

Sampler	Input parameters
MCMC	$N = 10000$, $\mathit{burnin} = 1000$, $\mathit{thin} = 1$
IMIS	$N_0 = 6000$, $B = 600$, $M = 3000$
SIR	$m = 50000$

Table 1: Input parameters for each posterior sampling algorithm.

Parameter	Real value	Sampler	2.5%	Median	97.5%	Mean	SD
β	5.000	SIR	3.613	7.874	7.874	6.543	1.677
		IMIS	3.728	5.091	6.859	5.158	0.801
		MCMC	3.736	5.141	6.970	5.201	0.819
		BCLT	3.562	4.985	6.351	4.963	0.716
Q	13.800	SIR	13.356	14.494	14.562	14.326	0.477
		IMIS	11.403	14.570	16.872	14.458	1.573
		MCMC	11.375	14.705	16.897	14.552	1.577
		BCLT	11.736	14.251	16.478	14.212	1.325
G	351.540	SIR	310.223	414.590	469.354	393.996	46.670
		IMIS	296.300	375.007	463.859	376.749	45.239
		MCMC	294.689	379.521	468.779	379.639	45.466
		BCLT	304.889	369.093	444.387	370.910	36.885
τ_N	1.000	SIR	0.957	0.957	1.735	1.129	0.297
		IMIS	0.984	1.283	1.738	1.302	0.192
		MCMC	0.993	1.289	1.723	1.308	0.188
		BCLT	0.963	1.263	1.662	1.275	0.179
τ_F	0.640	SIR	0.683	0.683	0.959	0.729	0.072
		IMIS	0.577	0.747	0.989	0.756	0.105
		MCMC	0.572	0.742	0.989	0.752	0.108
		BCLT	0.553	0.723	0.944	0.731	0.102
τ_{NF}	0.500	SIR	0.320	0.376	0.617	0.412	0.103
		IMIS	0.375	0.565	0.826	0.576	0.117
		MCMC	0.375	0.567	0.828	0.576	0.116
		BCLT	0.359	0.561	0.792	0.565	0.111

Table 2: Posterior summaries – Dependent model.

The argument `data` is a 3-column matrix such that the columns are time, exposure concentrations at the near field and at the far field, respectively. The argument `mcmc.control` is a list that contains the input parameters for the Metropolis-within-Gibbs algorithm. Similarly, there are control input arguments to BCLT, IMIS and SIR as well, which are `bclt.control`, `imis.control` and `sir.control`, respectively. More details about the arguments in `B2ZM` can be found in R using the line command `help(B2ZM)`.

As discussed in Section 3, the sampling algorithms require some input parameters. Table 1 presents the input parameters provided for each sampling algorithm in this example. The BCLT implemented in the **B2Z** package requires two input parameters: `m` and `sample_size`.

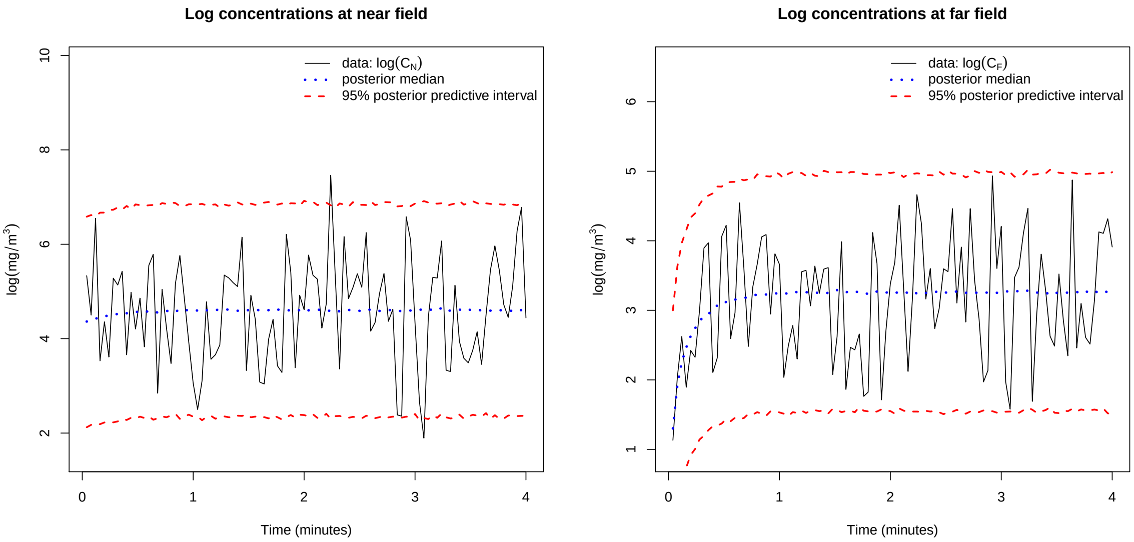

Figure 2: 95% posterior predictive intervals and posterior medians of the log exposure concentrations at the near and far fields over the observed period of time.

In particular, to estimate the posterior mode (needed in the BCLT), the function `nlminb` is used, which depends on the starting parameter values. The input `m` is the number of sampling values from the prior distributions of β, Q and G. Therefore, the vector among the `m` sampled with largest likelihood value is used as starting parameter values. The other input parameter `sample_size` is the size of the sample from the approximate posterior distribution of the parameters in the model according to the BCLT. We use `m = 8000` and `sample_size = 2000`.

Table 2 presents several posterior summaries for each parameter in the dependent model obtained by using the function `B2ZM` within the package **B2Z**. The IMIS and Metropolis-within-Gibbs algorithms provide similar estimates for the parameters in the model. In addition, the posterior means obtained by these algorithms fairly estimate the parameters in the model, except for β and G that were estimated using the SIR algorithm. The 95% credible intervals cover the true values of the parameters, except for τ_F when using the SIR algorithm.

In this example, the SIR algorithm samples poorly from the posterior distribution. In fact, the proportion of unique points in the sample is very low (0.062%), which explains the strange behavior in the standard deviation estimates. On the other hand, IMIS and Metropolis-within-Gibbs algorithms perform better. In particular, the IMIS has an expected fraction of unique points equaling 58.7% and the ESS for the Metropolis-within-Gibbs the acceptance rate is 51.27%.

The following figures are produced using the line command `plot(fit.depend)`, where the output `fit.depend` is an object from the Bayesian two-zone model fitted using the Metropolis-within-Gibbs algorithm. The analogous figures for the SIR and IMIS algorithms, and BCLT are not shown in this paper. However, they can be produced by running the example codes in the tutorial of **B2Z**.

Figure 2 shows the 95% posterior predictive intervals and the posterior medians of the log exposure concentrations at the near and far fields. These graphs help environmental researchers

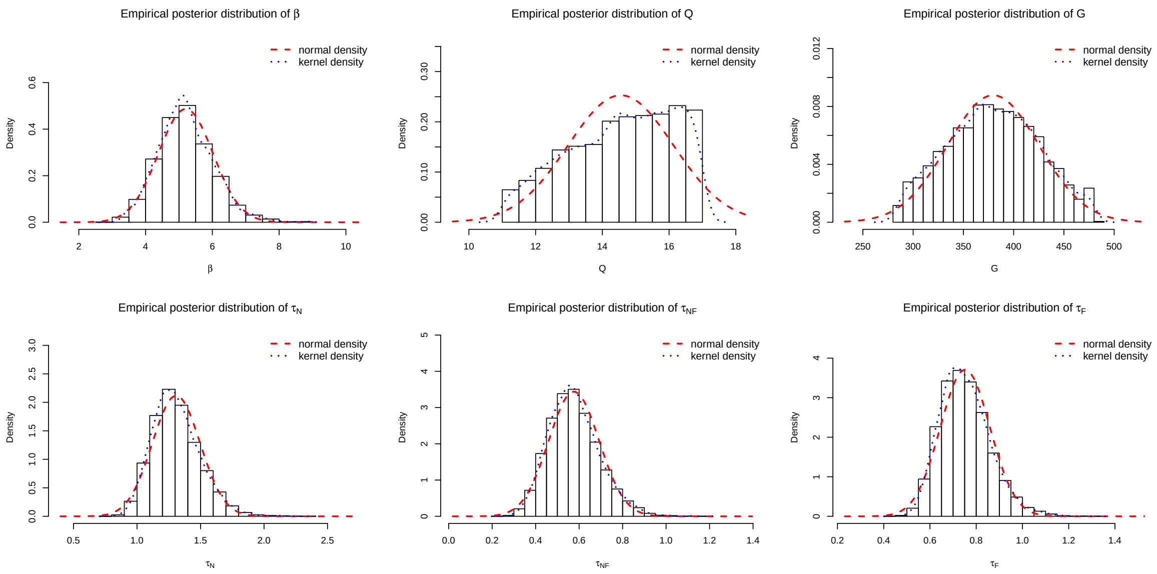

Figure 3: Empirical posterior distributions of the parameters in the dependent model.

know more about the range of the log exposure concentrations over the observed period of time in both fields. The solid lines in Figure 2 represent the observed log exposure concentrations.

Figure 3 shows the empirical posterior distributions of the parameters in the dependent model. Each empirical posterior distribution contains two curves:

- normal density centered at the estimated posterior mean and scaled by the estimated posterior standard deviation of the parameter;

- Gaussian kernel density curve.

Figure 4 shows the Metropolis history plot and ACF for the parameters in the model.

The Bayesian two-zone model fitting was done on a PC Intel Core Duo CPU P8600 with 2.40GHz and 4.00GB of Memory RAM. The computational time (in seconds) obtained for the SIR, IMIS, Metropolis-within-Gibbs and BCLT are 59.36, 133.30, 67.40, and 18.64, respectively. In this example, the computational time for the Metropolis-within-Gibbs also includes the time spent estimating the starting values and the covariance matrix needed for the proposal distribution.

B2Z can also interact with the package **coda**. For instance, Gelman and Rubin's convergence diagnostic can be computed very easily using the function `gelman.diag` provided by the package **coda**. To compute that measure we need to fit the model more than one time. In particular, we fit the model three times using Metropolis sampler and denote them by `fit.depend1`, `fit.depend2` and `fit.depend3`. The following code shows how to compute the Gelman and Rubin's convergence diagnostic in this example.

```
R> fit1 <- do.call(cbind, fit.depend1[c("Beta", "Q", "G")])
R> fit2 <- do.call(cbind, fit.depend2[c("Beta", "Q", "G")])
R> fit3 <- do.call(cbind, fit.depend3[c("Beta", "Q", "G")])
```

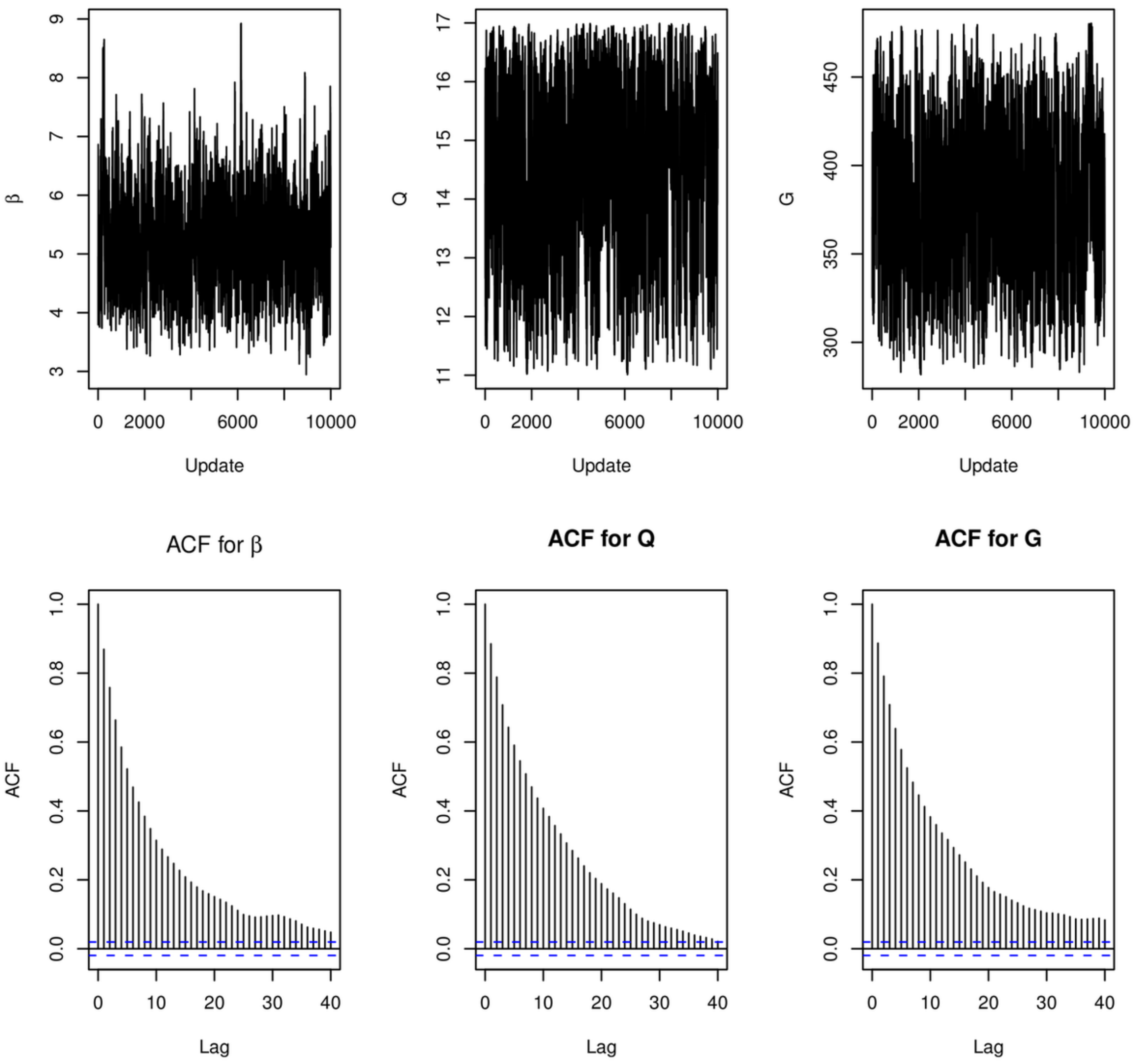

Figure 4: MCMC trace and ACF plots for β, Q and G.

```
R> x <- mcmc.list(list(mcmc(fit1), mcmc(fit2), mcmc(fit3)))
R> gelman.diag(x)

Potential scale reduction factors:

     Point est. 97.5% quantile
Beta       1.01           1.02
Q          1.00           1.01
G          1.00           1.01
```

```
Multivariate psrf

1.01
```

Since the values in the output above are close to 1, we conclude that there is no evidence that the chain does not converge. For further information regarding Gelman and Rubin's convergence diagnostic see Gelman and Rubin (1992). A multivariate version of Gelman and Rubin's diagnostic was proposed by Brooks and Gelman (1998).

4.2. Simulated data 2

Here we consider another simulated dataset with most parameters that generated exposure concentrations being the same as those in Section 4.1. The only difference is that now the measurement errors at the near and far field are considered independent. In particular, we set $\mathbf{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 0.64 \end{bmatrix}$. As in the previous section, we assume $\beta \sim \mathit{Unif}(0,10)$, $Q \sim \mathit{Unif}(11,17)$ and $G \sim \mathit{Unif}(281,482)$. However, now we fit the independent model and therefore we assume that $\tau_N \sim IG(5,4)$ and $\tau_F \sim IG(5,7)$.

The example code below shows how to specify the modeling information and the sampling algorithm desired using **B2Z**.

```
R> set.seed(2011)
R> fit.indep <- B2ZM(data = ex2, indep.model = TRUE,
+    priorBeta = "unif(0,10)", priorQ = "unif(11,17)",
+    priorG = "unif(281,482)", tauN.sh = 5, tauN.sc = 4, tauF.sh = 5,
+    tauF.sc = 7, VN = pi * 10^-3, VF = 3.8, sampler = "IMIS",
+    imis.control = list(N0 = 5000, B = 500, M = 3000, it.max = 16))
```

The input parameters used by the algorithms are the same as the ones presented in Table 1, except for the IMIS algorithm, which uses $N_0 = 5000$ and $B = 500$ in this section.

Table 3 presents posterior summaries for each parameter in the independent model. The parameters estimate using IMIS and Metropolis-within-Gibbs sampler are similar. Considering the parameters β,G and τ_F, the BCLT posterior means were closer to the true values than using the other algorithms.

The three samplers and the BCLT have very similar posterior summaries. All the 95% credibility intervals cover the true values of the parameters. Compared to the previous example, the performance of SIR is slightly better; the proportion of unique sampled points is 2.5%. The IMIS and Metropolis-within-Gibbs algorithms sample fairly well from the posterior distribution. As a matter of fact, the fraction of unique points for IMIS is 0.654 and the acceptance rate for the Metropolis-within-Gibbs is 49.23%.

The computational times (in seconds) for SIR, IMIS, Metropolis-within-Gibbs and BCLT were 29.50, 83.50, 41.01 and 8.05 respectively. Again, the computational time for Metropolis-within-Gibbs includes the time to estimate the covariance matrix of the proposal distribution. In the next section we illustrate an application of **B2Z** to a real experimental data set.

Parameter	Real value	Sampler	2.5%	Median	97.5%	Mean	SD
β	5.000	SIR	3.074	4.216	5.956	4.300	0.767
		IMIS	3.097	4.187	5.907	4.276	0.736
		MCMC	3.120	4.196	5.995	4.300	0.746
		BCLT	2.971	4.168	5.520	4.200	0.663
Q	13.800	SIR	11.772	14.828	16.913	14.818	1.382
		IMIS	11.954	14.988	16.919	14.835	1.406
		MCMC	12.030	15.058	16.904	14.900	1.370
		BCLT	12.176	14.915	16.651	14.763	1.252
G	351.540	SIR	281.791	331.052	408.878	332.522	36.291
		IMIS	284.285	328.712	410.566	333.464	34.857
		MCMC	285.367	329.867	408.828	334.313	33.837
		BCLT	294.470	333.618	412.503	339.556	32.246
τ_N	1.000	SIR	0.704	0.926	1.182	0.933	0.125
		IMIS	0.700	0.910	1.189	0.920	0.127
		MCMC	0.702	0.907	1.192	0.919	0.127
		BCLT	0.695	0.895	1.169	0.904	0.123
τ_F	0.640	SIR	0.570	0.706	0.962	0.733	0.107
		IMIS	0.569	0.728	0.953	0.735	0.098
		MCMC	0.568	0.731	0.961	0.740	0.102
		BCLT	0.557	0.721	0.945	0.729	0.098

Table 3: Posterior summaries – Independent model.

4.3. Experimental two-zone study

In this section we fit the Bayesian two-zone model to the data set used in the experimental two-zone study in Zhang *et al.* (2009). Here, exposure concentrations of Toluene over a period of time were observed. These measurements were made in four directions (east, west, north and south) on three horizontal parallel planes at 5 different distances ($10cm$, $15cm$, $20cm$, $30cm$, and $40cm$) from the contamination source, where the source was located on the middle plane and the exposure concentrations were measured every 5 seconds for at least 15 minutes in each location. Although combinations of factors such as presence of a worker's body, body movement and heat were also included in the experimental study, here we consider only the plain experimental data, i.e, the measurements that do not include any of those factors. A very detailed explanation of this experiment can be found in Zhang *et al.* (2009).

To illustrate **B2Z** using this real data set, we use the observed exposure concentrations on the middle plane and north direction. The measurements at 10 cm and 15 cm from the contamination source represent the exposure concentrations at the near and far fields, respectively. The volumes of the near and far fields are $\pi \times 10^{-3} m^3$ and $3.8m^3$, respectively. There are

140 observed time points equally spaced between 5 and 700 seconds.

We start with the dependent model. We let $\beta \sim \mathit{Unif}(0, 10)$, $Q \sim \mathit{Unif}(11, 17)$ and $G \sim \mathit{Unif}(281, 482)$. We also assume $\boldsymbol{\Sigma} \sim IW\left(\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, 4\right)$. To fit this model using the IMIS sampler, we use the following line command:

```
R> fit.imis <- B2ZM(data = real.data, priorBeta = "unif(0,10)",
+    indep.model = FALSE, priorQ ="unif(11,17)", priorG = "unif(281,482)",
+    S = diag(10,2), v = 4, VN = pi * 10^-3, VF = 3.8, sampler ="IMIS",
+    imis.control = list( N0 = 6000, B = 600, M = 3000, it.max = 16))
```

Now we fit the Bayesian two-zone model using the Metropolis sampler. However, unlike the previous two sections, we provide the covariance matrix in the proposal distribution for the Metropolis-within-Gibbs algorithm. To do this, we use the output `imis.control` to form a guess for such a matrix. The following line commands show how this can be done:

```
R> initial <-  summary(fit.imis)$summary[, "Mean"][1:3]
R> prop.matrix <- summary(fit.imis)$PostCovMat[1:3, 1:3]
```

Therefore, defining the covariance matrix for the proposal distribution in the function `B2ZM` is very straightforward, as given in the code below:

```
R> fit.mcmc <- B2ZM(data = real.data, priorBeta = "unif(0,10)",
+    indep.model = FALSE, priorQ = "unif(11,17)", priorG = "unif(281,482)",
+    S = diag(10,2), v = 4, VN = pi * 10^-3, VF = 3.8, sampler = "MCMC",
+    mcmc.control = list(initial = initial, Sigma.Cand = prop.matrix,
+      NUpd = 10000, burnin = 1000, lag = 1))
```

Parameter	Sampler	2.5%	Median	97.5%	Mean	SD
β	IMIS	8.131	9.320	9.840	9.238	0.467
	MCMC	8.090	9.287	9.939	9.228	0.498
Q	IMIS	11.002	11.043	11.200	11.058	0.056
	MCMC	11.001	11.039	11.221	11.058	0.061
G	IMIS	473.059	480.086	481.873	479.307	2.592
	MCMC	473.583	480.276	481.905	479.554	2.396
τ_N	IMIS	0.065	0.080	0.102	0.081	0.010
	MCMC	0.065	0.081	0.104	0.082	0.010
τ_F	IMIS	0.269	0.340	0.446	0.343	0.045
	MCMC	0.270	0.339	0.435	0.342	0.042
τ_{NF}	IMIS	−0.045	0.000	0.036	−0.001	0.021
	MCMC	−0.047	0.000	0.037	−0.001	0.021

Table 4: Posterior summaries – Experimental data set – Dependent model.

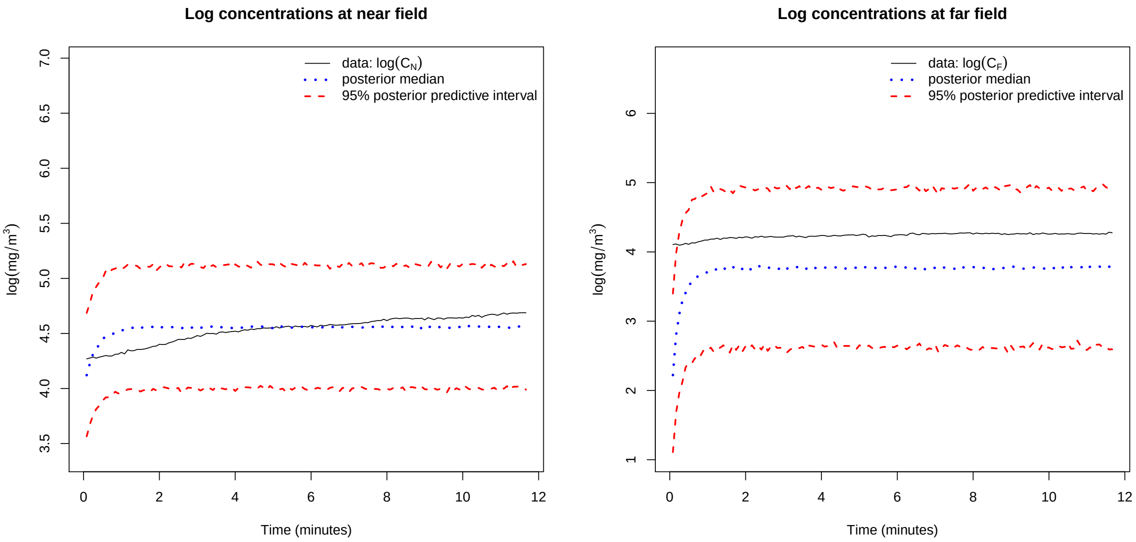

Figure 5: 95% posterior predictive intervals and posterior medians of the log exposure concentrations at the near and far fields over the observed period of time – Real experimental data set.

Parameter	2.5%	Median	97.5%	Mean	SD
β	8.634	9.358	9.865	9.333	0.325
Q	11.004	11.038	11.191	11.052	0.047
G	473.647	480.118	481.685	479.481	2.192
τ_N	0.051	0.063	0.080	0.064	0.008
τ_F	0.283	0.348	0.440	0.352	0.042

Table 5: Posterior summaries – Experimental data set – Independent model.

The estimates for the parameters using IMIS and Metropolis-within-Gibbs are presented in Table 4.

Figure 5 shows the 95% posterior predictive intervals and the posterior medians of the log exposure concentrations at the near and far fields using the Metropolis-within-Gibbs sampler. We discover that the posterior medians do not predict the log exposure concentrations very well, especially for the far field.

Table 4 reveals that the Metropolis and IMIS algorithms yield similar estimates. The DICs found using IMIS and Metropolis algorithms are 140.224 and 140.816, respectively. We also fit the independent model, for which we assume $\tau_N \sim IG(5, 4)$ and $\tau_F \sim IG(5, 7)$. Since in the previous examples we noticed that the algorithms in general have similar estimates, we only fit the independent Bayesian two-zone model using the IMIS algorithm. Posterior summaries of the parameters in the independent model are presented in Table 5.

The DIC for the fitted independent model is 115.392, which indicates that the independent model fits the data better than the dependent model. However, the estimates in Table 5 are not substantially different from those in Table 4.

5. Discussion

In this paper we have introduced our user-friendly R package **B2Z**. We have described the underlying models and a suite of algorithms to perform Bayesian inference on the two-zone models in occupational hygiene (e.g., Zhang *et al.* 2009). In particular, we have demonstrated the main function called `B2ZM`, where the output from this function is a valid input for the functions `summary` and `plot`. Currently **B2Z** implements three different samplers: SIR, IMIS, and the MCMC. In addition, the package also offers approximate Bayesian inference using the Bayesian central limit theorem. Our illustrative examples show that IMIS, MCMC and BCLT obtain similar posterior summaries. The SIR's performance was somewhat inferior, but it is easier to implement and can be useful as an initial tool for exploring approximate posteriors. Our examples also show that the Bayesian two-zone model can be fitted within a reasonable time (ranging from 8 to 140 seconds). In particular, IMIS is the slowest algorithm while BCLT has the fastest computational time. In ongoing work we focus upon including new features: adaptive MCMC samplers, maximum likelihood estimators, and a function called `predict` where users can define time intervals over which concentration predictions are sought. The **B2Z** 1.4 is already available for download from the Comprehensive R Archive Network at http://CRAN.R-project.org/package=B2Z.

References

Bishop CM (2006). *Pattern Recognition and Machine Learning.* Springer-Verlag, New York.

Brooks SP, Gelman A (1998). "General Methods for Monitoring Convergence of Iterative Simulations." *Journal of Computational and Graphical Statistics*, **7**(4), 434–455.

Carlin BP, Louis TA (2008). *Bayesian Methods for Data Analysis.* 3rd edition. Chapman & Hall/CRC, New York.

Cherrie JW, Gelman A (1999). "The Effect of Room Size and General Ventilation on Relationship Between Near and Far-Field Concentrations." *Applied Occupational and Environmental Hygiene*, **14**, 539–546.

Dijk HKV, Hop JP, Louter AS (1987). "An Algorithm for the Computation of Posterior Moments and Densities Using Simple Importance Sampling." *The Statistician*, **36**, 83–90.

Gay DM (1990). "Usage Summary for Selected Optimization Routines." *Technical Report 153*, AT & T Bell Laboratories, Department of Computing Science. URL http://netlib.bell-labs.com/cm/cs/cstr/153.pdf.

Gelfand AE, Smith AFM (1990). "Sampling-Based Approaches to Calculating Marginal Densities." *Journal of the American Statistical Association*, **85**(410), 398–409.

Gelman A, Carlin JB, Stern HS, Rubin DB (2003). *Bayesian Data Analysis.* 2nd edition. Chapman & Hall/CRC.

Gelman A, Rubin DB (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472.

Geman S, Geman D (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(6), 721–741.

Gilbert P (2011). ***numDeriv**: Accurate Numerical Derivatives.* R package version 2010.11-1, URL http://CRAN.R-project.org/package=numDeriv.

Hastings WK (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika*, **57**(1), 97–109.

Laub AJ (2005). *Matrix Analysis for Scientists and Engineers.* Society for Industrial and Applied Mathematics, Philadelphia.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953). "Equation of State Calculations by Fast Computing Machines." *Journal Chemical Physics*, **21**(6), 1087–1092.

Nicas M (1996). "Estimating Exposure Intensity in an Imperfectly Mixed Room." *American Industrial Hygiene Association Journal*, **57**, 542–550.

Nicas M, Jayjock M (2002). "Uncertainty in Exposure Estimates Made by Modeling Versus Monitoring." *American Industrial Hygiene Association Journal*, **63**, 275–283.

Nicas M, Miller SL (1999). "A Multi-Zone Model Evaluation of the Efficacy of Upper-Room Air Ultraviolet Germicidal Irradiation." *Applied Occupational and Environmental Hygiene*, **14**, 317–328.

Plummer M, Best N, Cowles K, 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/.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Raftery AE, Bao L (2010). "Estimating and Projecting Trends in HIV/AIDS Generalized Epidemics Using Incremental Mixture Importance Sampling." *Biometrics*, **66**, 1162–1173.

Ramachandran G (2005). *Occupational Exposure Assessment for Air Contaminants.* CRC Press, Taylor & Francis Group.

Robert CP, Casella G (2004). *Monte Carlo Statistical Methods.* Springer-Verlag, New York.

Rubin DB (1987). "The Calculation of Posterior Distributions by Data Augmentation: Comment: A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputations When Fractions of Missing Information Are Modest: The SIR Algorithm." *Journal of the American Statistical Association*, **82**(398), 543–546.

Rubin DB (1988). "Using the SIR Algorithm to Simulate Posterior Distributions." In JM Bernardo, MH Degroot, DV Lindley, AFM Smith (eds.), *Bayesian Statistics 3*. Oxford University Press.

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

Steele RJ, Raftery AE, Emond MJ (2006). "Computing Normalizing Constants for Finite Mixture Models via Incremental Mixture Importance Sampling (IMIS)." *Journal of Computational and Graphical Statistics*, **15**(3), 712–734.

Zhang Y, Banerjee S, Yang R, Lungu C, Ramachandran G (2009). "Bayesian Modeling of Exposure and Air Flow Using Two-Zone Models." *The Annals of Occupational Hygiene*, **53**(4), 409–424.

Affiliation:

João Vitor Dias Monteiro, Sudipto Banerjee
School of Public Health
Division of Biostatistics
University of Minnesota
A450 Mayo Building
420 Delaware St. S.E.
Minneapolis, MN 55455, United States of America
E-mail: monte092@umn.edu, baner009@umn.edu

Gurumurthy Ramachandran
Division of Environmental Health Sciences
University of Minnesota
420 Delaware Street SE, MMC 807
Minneapolis, MN 55455, United States of America
E-mail: ramac002@umn.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 43, Issue 2 *Submitted:* 2010-06-09
July 2011 *Accepted:* 2011-05-12