Research Article

Statistics in Medicine

Received XXXX

(www.interscience.wiley.com) DOI: 10.1002/sim.0000

Reducing Monte Carlo error in the Bayesian estimation of risk ratios using log-binomial regression models

D. Salmerón,[abcd]* J. A. Cano[e] and M. D. Chirlaque[abc]

In cohort studies binary outcomes are very often analyzed by logistic regression. However, it is well-known that when the goal is to estimate a risk ratio, the logistic regression is inappropriate if the outcome is common. In these cases, a log-binomial regression model is preferable. On the other hand, the estimation of the regression coefficients of the log-binomial model is difficult due to the constraints that must be imposed on these coefficients. Bayesian methods allow a straightforward approach for log-binomial regression models, produce smaller mean squared errors in the estimation of risk ratios than the frequentist methods, and the posterior inferences can be obtained using the software WinBUGS. However, Markov chain Monte Carlo (MCMC) methods implemented in WinBUGS can lead to large Monte Carlo errors in the approximations to the posterior inferences since they produce correlated simulations and the accuracy of the approximations are inversely related to this correlation. To reduce correlation and to improve accuracy, we propose a reparameterization based on a Poisson model and a sampling algorithm coded in R. Copyright © xxxx John Wiley & Sons, Ltd.

Keywords: binomial regression models; Markov chain Monte Carlo; Monte Carlo error; Bayesian inference

1. Introduction

The odds ratio is a measure of association widely used in Epidemiology that can be estimated using logistic regression. However, when one wants to communicate a risk ratio, the logistic regression is not recommended if the outcome is common [1-6]. If one wants to estimate the adjusted risk ratio, a log-binomial model is preferable to a logistic model. The

[a] *CIBER Epidemiología y Salud Pública (CIBERESP), Spain*
[b] *Servicio de Epidemiología, Consejería de Sanidad y Política Social, Ronda de Levante 11, E30008-Murcia, Spain*
[c] *Instituto Murciano de Investigación Biosanitaria Virgen de la Arrixaca, IMIB*
[d] *Departamento de Ciencias Sociosanitarias, Universidad de Murcia, Spain*
[e] *Departamento de Estadística e Investigación Operativa, Universidad de Murcia, E30100-Espinardo, Spain*
* *Correspondence to: Diego Salmerón. CIBER Epidemiología y Salud Pública (CIBERESP), Spain. E-mail: dsm@um.es*

Contract/grant sponsor: This research was supported by the Séneca Foundation Programme for the Generation of Excellence Scientific Knowledge under Project 15220/PI/10.

log-binomial model assumes that the distribution of the outcome y_i is the Bernoulli distribution

$$y_i \sim Ber(p_i), \;\; \log p_i = x_i\beta, \;\; i \in \mathbb{N}_n = \{1, ..., n\}, \tag{1}$$

where,

$$x_i\beta = (x_{i1}, x_{i2}, ..., x_{ik})(\beta_1, ..., \beta_k)^T = x_{i1}\beta_1 + \cdots + x_{ik}\beta_k,$$

and x_i includes variables denoting exposures, confounders, predictors and product terms. Usually $x_{i1} = 1$ and therefore β_1 is the intercept. The parameters $\exp(\beta_j)$ are interpreted as adjusted risk ratios, whether the outcome is common or rare.

Since $p_i = \exp(x_i\beta) \in (0, 1)$, we have to impose the constraints $x_i\beta < 0$, $i \in \mathbb{N}_n$, on the values of β, which complicates its maximum likelihood estimation. Zou [7] and Spiegelman and Hertzmark [4] have suggested the use of a Poisson model without the constraints, that is,

$$y_i \sim Poisson(\mu_i), \;\; \log \mu_i = x_i\beta, \;\; i \in \mathbb{N}_n, \tag{2}$$

to approximate the log-binomial maximum likelihood estimator, and they consider a robust sandwich variance estimator to estimate the standard errors. Model (2) can be fitted with standard statistical packages. Nevertheless, if $\hat{\beta}$ is the estimate obtained fitting the Poisson model (2), then $x_i\hat{\beta}$ can be greater than zero. On the other hand, Petersen and Deddens [5, 6] have proposed a different approximation using an *expanded dataset* that contains $(c - 1)$ copies of the original data and one copy of the original data with the dependent variable values interchanged (1's changed to 0's and 0's changed to 1's). As c becomes large, the maximum likelihood estimator for this modified data set approaches the maximum likelihood estimator for the original data set. Savu *et al*. [8] have shown the existence and the uniqueness of the estimator produced by this method.

In this article we consider the Bayesian analysis of the log-binomial regression model (1). In this context, Chu and Cole [9] have proposed to incorporate the constraints $x_i\beta < 0$, $i \in \mathbb{N}_n$, as part of the likelihood function using the indicator function:

$$\prod_{i=1}^{n} \exp(x_i\beta)^{y_i}(1 - \exp(x_i\beta))^{1-y_i} I(x_i\beta < 0),$$

where $I(x_i\beta < 0) = 1$ if $x_i\beta < 0$, and 0 otherwise. They have shown that the Bayesian approach provides estimates similar to the maximum likelihood estimates, produces smaller mean squared errors in the estimation of risk ratios, and posterior computations can be carried out using the Markov chain Monte Carlo (MCMC) methods implemented in WinBUGS [10]. However, MCMC methods can lead to another error, that is, MCMC methods produce a sample from the posterior distribution and approximate the posterior inferences using this sample, and therefore, every time we obtain a sample from the posterior distribution, the resulting approximations are different and Monte Carlo errors are present. That is, the sample generated by MCMC methods is a Markov chain that consists of correlated simulations from the posterior distribution, and the Monte Carlo error in the results is related to this correlation, affecting the numerical accuracy of the approximations [11].

As we show in this article, WinBUGS can lead to large Monte Carlo errors in the approximations. Therefore a very large number of iterations (the length of the chain) would be needed to obtain a desirable accuracy. Furthermore, WinBUGS does not take into account the constraints in an efficient way to produce the Markov chain. The reason is that to generate a value of β, WinBUGS has to propose k random values without taken into account the constraints, and then WinBUGS has to evaluate the constraints. Every new proposed random value is not accepted if the corresponding n constraints are not satisfied. It would be better if the proposed value satisfied the constraints from the time it is proposed since the acceptance rate also affects the numerical accuracy of the approximations.

In this work we overcome these two drawbacks using a specific reparameterization. This sampling algorithm has been implemented using the statistical package R [12], and it is very easy to use because researchers only need to fit the Poisson model (2) using the glm function. We establish the reparameterization in section 2, and in section 3 we develop a specific

Gibbs sampler. The method is applied in section 4 with data from the Murcia population-based cancer registry. In section 5 we compare our method with WinBUGS using two real examples. A simulation study is performed in section 6, and the conclusions are stated in section 7.

2. Reparameterization to improve the accuracy

To reduce correlation we propose a reparameterization replacing the original parameter β with a new parameter θ whose covariance matrix given the data is approximately the identity matrix. It is well-known that this approach can improve the performance of MCMC methods like the Gibbs sampler [13]. This reparameterization may be obtained using the estimated covariance matrix of the maximum likelihood estimator of β in model (1). However, very often neither the maximum likelihood estimates nor the estimated covariance matrix can be calculated for log-binomial regression models. To avoid this drawback we propose a reparameterization based on a Poisson model [7] as follows.

Let S be the estimated covariance matrix of the maximum likelihood estimator $\hat{\beta}$ obtained fitting the Poisson model (2) and let L be a matrix such that $S = L^T L$. The vector $\hat{\beta}$ and the matrix S can be easily obtained with the statistical package R [12] and the glm function, the matrix L is computed using the R function chol: $L = \mathrm{chol}(S)$, and the matrix L^T denotes the transpose of L. The reparameterization we propose is

$$\beta = L^T\theta, \ \ \text{with} \ \ \theta = (\theta_1, \ldots, \theta_k)^T.$$

The likelihood function associated with the log-binomial regression model (1) is $f(\mathbf{y} \mid \beta) = \prod_{i=1}^n p_i^{y_i}(1-p_i)^{1-y_i}$, where $\mathbf{y} = (y_1, \ldots, y_n)$, $p_i = \exp(x_i\beta)$ and $x_i\beta < 0$, $i \in \mathbb{N}_n$. If $\pi(\beta)$ is the prior distribution, then the posterior distribution is $\pi(\beta \mid \mathbf{y}) \propto \pi(\beta)f(\mathbf{y} \mid \beta)$ and hence, given the data $\mathbf{y}$, the distribution of θ is $\pi(\theta \mid \mathbf{y}) \propto \pi(L^T\theta)\prod_{i=1}^n p_i^{y_i}(1-p_i)^{1-y_i}$, $\theta \in \Theta$, where now $p_i = \exp(z_i\theta)$, $z_i = x_i L^T$, $i \in \mathbb{N}_n$, and $\Theta = \{\theta \in \mathbb{R}^k;\ z_i\theta < 0,\ \ i \in \mathbb{N}_n\}$. Note that

$$L^T L = S \approx \mathrm{Cov}(\beta \mid \mathbf{y}) = L^T \mathrm{Cov}(\theta \mid \mathbf{y}) L,$$

and therefore $\mathrm{Cov}(\theta \mid \mathbf{y})$ is approximately the identity matrix, that is, $(\theta_1, \ldots, \theta_k)$ are approximately uncorrelated, given the data. Hence a Gibbs sampler with target $\pi(\theta \mid \mathbf{y})$ might get better convergence than a Gibbs sampler used to simulate from $\pi(\beta \mid \mathbf{y})$.

3. A specific Gibbs sampler to simute from $\pi(\theta \mid \mathbf{y})$

As we have pointed out above, WinBUGS does not take into account the constraints in an efficient way to produce the Markov chain. To overcome this drawback we consider a specific Gibbs sampler [13] to simulate from $\pi(\theta \mid \mathbf{y})$. Given the value of the parameter of the chain at iteration t,

$$(\theta_1(t), \theta_2(t), \ldots, \theta_k(t)),$$

the next value

$$(\theta_1(t+1), \theta_2(t+1), \ldots, \theta_k(t+1))$$

is generated sequentially updating each coordinate conditionally on the present values of the remaining coordinates and the data. More concretely, suppose that we have generated $\theta_1(t+1), \ldots, \theta_{j-1}(t+1)$. To generate $\theta_j(t+1)$ we carry out a Metropolis-Hastings step with target

$$\pi(\theta_j \mid \theta_1(t+1), \theta_2(t+1), \ldots, \theta_{j-1}(t+1), \theta_{j+1}(t), \ldots, \theta_k(t), \mathbf{y}).$$

To perform the Metropolis-Hastings step we consider

$$\theta = (\theta_1(t+1), \theta_2(t+1), \ldots, \theta_{j-1}(t+1), \theta_j(t), \theta_{j+1}(t), \ldots, \theta_k(t))$$

Then we simulate θ'_j from a *proposal* distribution $\mathcal{C}(\theta'_j)$, and we define θ' as

$$\theta' = (\theta_1(t+1), \theta_2(t+1), \ldots, \theta_{j-1}(t+1), \theta'_j, \theta_{j+1}(t), \ldots, \theta_k(t)).$$

Finally, we simulate v from the uniform distribution $U(0,1)$ and then

$$\theta_j(t+1) = \begin{cases} \theta'_j & \text{if } v < \rho \\ \theta_j(t) & \text{if } v \geq \rho \end{cases}$$

where,

$$\rho = \min\left(1, \frac{\pi(\theta' \mid \mathbf{y})\mathcal{C}(\theta_j(t))}{\pi(\theta \mid \mathbf{y})\mathcal{C}(\theta'_j)}\right).$$

The proposal distribution $\mathcal{C}(\theta'_j)$ is chosen as follows. Note that $\mathrm{Cov}(\theta \mid \mathbf{y})$ is approximately the identity matrix, $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_k)^T = L^{-T}\hat{\beta}$ approximates the mode of $\pi(\theta \mid \mathbf{y})$ assuming a non-informative prior for θ, and that given $\theta_1(t+1)$, $\theta_2(t+1)$, $\ldots$, $\theta_{j-1}(t+1)$, $\theta_{j+1}(t)$, $\ldots$, $\theta_k(t)$ and $\mathbf{y}$, θ_j is restricted to lie in the interval $\Theta_j = (a_j, b_j)$ (see Appendix). Therefore, the normal distribution $N(\hat{\theta}_j, 1)$ restricted to $\Theta_j = (a_j, b_j)$ may be an appropriate proposal distribution to fulfill the Metropolis-Hastings step. On the other hand, to simulate from a truncated normal distribution we need the evaluation of the normal cumulative distribution function and of its inverse, which can increase the computational time. Instead of the truncated normal distribution, we propose the Cauchy distribution with location $\hat{\theta}_j$ and scale 1 restricted to Θ_j. The density of this distribution is

$$\mathcal{C}(\theta'_j) \propto \frac{1}{(1 + (\theta'_j - \hat{\theta}_j)^2)}, \quad \theta'_j \in \Theta_j.$$

To simulate θ'_j from this proposal distribution, we simulate $u \sim U(0,1)$ and compute

$$\theta'_j = \hat{\theta}_j - \tan\Big((u-1)\arctan(a_j - \hat{\theta}_j) + u\arctan(\hat{\theta}_j - b_j)\Big).$$

Finally, all the coordinates are generated in such a way that $(\theta_1(t+1), \theta_2(t+1), \ldots, \theta_k(t+1))$ automatically satisfies the n constraints. The proposed MCMC algorithm has been implemented with the statistical package R [12] and it is provided in the Appendix.

4. Survival of cancer patients using data from population-based cancer registries

The extent to which several factors, such as age, stage, sex or country, affect overall survival of cancer patients is of great importance for the assessment of prognosis and patient care ([14, 15]).

To illustrate our method we used the cases of colon cancer diagnosed during the period 1995-2006 in ages 55-79 years, provided by the Murcia population-based cancer registry, which is member of the European Network of Cancer Registries. We studied the effect of sex, age, year of diagnosis, histology and stage, on the probability of being alive five years after of being diagnosed with colon cancer, having survived one year. All patients ($N = 2,949$) were followed up to December 31st, 2011; 2,000 (67.8%) were alive five years after diagnosis, and 949 (32.2%) died before.

The outcome is $y = 1$ if the patient was alive 5 years after diagnosis, and 0 if the patient died before. The log-binomial regression model is

$$\log p_i = x_i\beta = \beta_1 + \beta_2\mathrm{Female}_i + \beta_3\mathrm{StageI}_i + \beta_4\mathrm{StageII}_i + \beta_5\mathrm{StageIII}_i + \beta_6\mathrm{StageUnknown}_i +$$

$$\beta_7\mathrm{Age55}_i + \beta_8\mathrm{Age60}_i + \beta_9\mathrm{Age65}_i + \beta_{10}\mathrm{Age70}_i + (\beta_{11}, \ldots, \beta_{21})(\mathrm{Year2}_i, \ldots, \mathrm{Year12}_i)^T + \beta_{22}\mathrm{Adeno}_i,$$

$i = 1, \ldots, 2,949$. The variables Female, StageI, StageII, StageIII, StageUnknown, Age55, ..., Age70, Year2, ..., Year12 and Adeno are indicator variables, that is, Female=1 for females and 0 for males, StageI=1 for cancers in stage I and 0 otherwise, Age55=1 if the age at diagnosis was in the interval [55,60) and 0 otherwise, Year2=1 if the year of the diagnosis was 1996 and 0 otherwise, and Adeno=1 if the histological group was adenocarcinoma (excluding mucinous) and 0 otherwise. The reference categories are shown in Table 1. For example, the parameter $\exp(\beta_3)$ is a risk ratio: the probability of being alive five years after diagnosis (having survived one year) among patients with the cancer in stage I, divided by that probability among patients with the cancer in stage IV.

We simulated a chain of 30,000 iterations using WinBUGS and a chain of 30,000 iterations using our method. The first 20,000 values of each chain were discarded. Then we approximated the posterior mean and the credible interval (based on the posterior quantiles $Q(0.025)$ and $Q(0.975)$) of $RR_j = \exp(\beta_j)$, $j = 1, \ldots, 22$, using these chains. The prior distribution was $\pi(\beta) \propto 1$ over the region defined by the constraints.

Table 1 shows the estimates (for the variables sex, age group and stage) obtained fitting the logistic regression model $\mathrm{logit}\, p_i = x_i\beta$, $i = 1, \ldots, 2,949$, and fitting the log-binomial regression model using the proposed method and using WinBUGS. The estimates of the odds ratios were greater than the estimates of the risk ratios. On the other hand, the approximations obtained fitting the log-binomial model with the proposed method were very similar to those obtained with WinBUGS for all the parameters except for stage. Figure 1 shows the trace plots and autocorrelation functions obtained from the proposed algorithm (first row) and from WinBUGS (second row) for the parameter $\exp(\beta_3)$ associated with stage I. Using the proposed method, autocorrelation values dissipate rapidly, contrary to what happened using WinBUGS. This and the trace plots indicate that the proposed method converges faster than WinBUGS towards the posterior distribution.

5. Comparison with WinBUGS

In this section we present two examples to compare the proposed method with WinBUGS. For each example we proceeded in the following way to approximate the posterior distribution of $\exp(\beta_j)$, $j = 1, \ldots, k$. We simulated 500 different values of β that satisfy the constraints. These values were used to initialize the Markov chains. For each initial value we simulated a chain of 10,000 iterations using WinBUGS and a chain of 10,000 iterations using our method. The first 500 values of each chain were discarded. Then we approximated the posterior mean, the posterior standard deviation and the posterior quantiles $Q(0.025)$ and $Q(0.975)$ of $\exp(\beta_j)$, $j = 1, \ldots, k$, using the 500 chains obtained from WinBUGS and the 500 chains obtained with our method. Therefore, for each summary statistics of the posterior distribution of $\exp(\beta_j)$, $j = 1, \ldots, k$, we obtained 500 approximations using WinBUGS and 500 approximations using our method. The prior distribution was $\pi(\beta) \propto 1$ over the region defined by the constraints. To assess the stability of our method compared with that of WinBUGS, we show the Monte Carlo error of each method using boxplots of these approximations. Also, we show the effective sample size [16] for the parameter $\exp(\beta_j)$, that is, the number of independent draws from the posterior of $\exp(\beta_j)$ that would give a Monte Carlo estimate of the posterior mean of $\exp(\beta_j)$ with the same level of precision as the estimate given by the Markov chain. We approximated the posterior distribution using our method with a chain of 1,000,000 iterations and the first 50,000 values of the chain were discarded.

5.1. *Breast cancer mortality*

We considered the data on the relation between receptor level and stage to 5-year survival in a cohort of 192 women with breast cancer discussed in Greenland [3]. In this example the percentage of deaths was 28.13%.

The Gelman and Rubin statistics indicated for both methods that the Markov chains tended to converge within 9,500 iterations (the potential scale reduction factors were less or equal than 1.03). However, boxplots of the approximations obtained with the proposed method and WinBUGS (see Figure 2) show that our method resulted in smaller between chain variability in the summary statistics (Table 2). We do not display the boxplots for the parameter $\exp(\beta_1)$ because they had a very different scale to the others, but they exhibit the same behaviour. The mean effective sample sizes obtained with our method were higher than those obtained with WinBUGS, see Table 3. These results show that our method converges faster, reduces correlation and is more stable than WinBUGS (regarding the computational speed, our algorithm and WinBUGS lasted 7 seconds to generate a chain).

To carry out the Metropolis-Hastings step, the truncated normal distribution can be used instead of the truncated Cauchy distribution. We have explored this option with the breast cancer example as follows. To simulate from the truncated normal distribution we have used the classical inversion technique. The two resulting methods have been used 1,000 times with chains of length 10,000 (discarding the first 500 iterations) and we have meausured the efficiency Eff of each method. The mean (standard deviation) of the 1,000 values of Eff for the parameters $\exp(\beta_j)$, $j = 1, \ldots, 4$, were 931.7(44.7), 724.8(40.8), 927.6(47.2), and 799.9(47.6), respectively, when we used the truncated Cauchy proposal, while they were 850.6(115.3), 771.2(43.4), 484.7(168.3), and 442.6(143.4) when we used the truncated normal one. In this example the truncated Cauchy distribution worked better than the truncated normal distribution.

5.2. *Low birth weight*

We used the data [17] from a 1986 cohort study conducted at the Baystate Medical Center, Springfield Massachusetts. The study was designed to identify risk factors associated with an increased risk of low birth weight (weighing less than 2,500 grams). Data were collected from 189 pregnant women, 59 of whom had low birth weight infants. We studied the association between the low birth weight and uterine irritability (ui: yes/no), smoking status during pregnancy (smoke: yes/no), mother's race (race: white, black, other), previous premature labours (ptl>0: yes/no), and mother's age (age: ≤ 18, (18,20], (20,25], (25,30] and > 30).

Figure 3 shows the boxplots of the approximations obtained with the proposed method and WinBUGS. The boxplots show that our method resulted in smaller between chain variability in the summary statistics (Table 4). We do not display the boxplots for the parameters $\exp(\beta_1)$ and $\exp(\beta_9)$ because they had a very different scale to the others, but they show the same behaviour. Mean effective sample sizes obtained with the proposed method were higher than those obtained with WinBUGS, see Table 5, reflecting reduction in the correlation. Again, the results show that our method converges faster and is more stable than WinBUGS (regarding the computational speed, our algorithm and WinBUGS lasted 20 seconds to generate a chain). On the other hand, the Gelman and Rubin statistics indicated for both methods that the Markov chains tend to converge within 9,500 iterations (the potential scale reduction factors were less or equal than 1.02).

6. Simulation study

To explore the efficiency of the proposed method compared to WinBUGS, we performed a simulation study as follows. We generated data from the log-binomial model

$$\log p_i = x_{i1}\beta_1 + x_{i2}\beta_2 + x_{i3}\beta_3, \quad i = 1, \ldots, 400,$$

where $x_{i1} = 1$, $x_{ij} = F(z_{ij})$, $j = 2, 3$, F is the distribution function of the standard normal distribution, and (z_{i2}, z_{i3}) was generated from a bivariate standard normal distribution with a correlation coefficient of -0.5. For each set of true values of the parameters in Table 6, we generated 1,000 datasets, $\{(y_i, x_i) : i = 1, \ldots, 400\}$, and for each dataset we fitted the model using our algorithm with a chain of 5,000 iterations discarding the first 500 values of the chain. The prior distribution was $\pi(\beta) \propto 1$ over the region defined by the constraints. For each chain we obtained the effective sample size, the computational time, and the approximate 95% credible interval based on the posterior quantiles $Q(0.025)$ and $Q(0.975)$. This simulation study was replicated using WinBUGS also with chains of 5,000 iterations discarding the first 500 values of each chain. The efficiency (Eff) of each method was measured as the effective sample size for each parameter $\exp(\beta_j)$ divided by the computational time required to obtain a chain.

Table 6 shows the mean and the standard deviation of the 1,000 values of Eff for each set of parameters, the percentage of times that the 95% credible interval covered the true value of the parameter, and the average length of the 95% posterior credible intervals. The mean efficiency of our method was always higher than the mean efficiency of WinBUGS. The coverage percentages provided by both methods were close to the nominal level. However, the coverage percentages obtained in this simulation study were overall higher using the proposed method. The absolute difference between the coverage percentages and the value 95 ranged from 0.1 to 1.1, from 0.1 to 1.6 and from 0.2 to 1.3, for $\exp(\beta_1)$, $\exp(\beta_2)$ and $\exp(\beta_3)$ respectively, when we used the proposed method. Using WinBUGS, the absolute difference ranged from 0.1 to 2.9, from 0.1 to 2.0 and from 0.6 to 2.7, for $\exp(\beta_1)$, $\exp(\beta_2)$ and $\exp(\beta_3)$ respectively. On the other hand, the average lengths of the posterior credible intervals were very similar.

7. Conclusions

Despite recent efforts made by several authors, logistic regression is still frequently used in cohort studies and clinical trials with equal follow-up times, even if one wants to communicate a risk ratio. It is well-known that the more frequent the outcome is the more the odds ratio overestimates the risk ratio when it is greater than 1 (or underestimates it if it is less than 1). If one wants to estimate an adjusted risk ratio, the log-binomial model is preferable to the logistic one but the constrained parameter space makes difficult to find the maximum likelihood estimate. Bayesian methods implemented with WinBUGS can work with a constrained parameter space in a natural way. Moreover, Chu and Cole[9] have shown that Bayesian methods produce smaller mean squared errors than likelihood based methods. However, WinBUGS can lead to large Monte Carlo errors. To avoid this drawback, we have proposed an efficient MCMC algorithm to estimate risk ratios from a Bayesian point of view using log-binomial regression models. Our method is based on two strategies: first, a reparameterization based on a Poisson model, and second, a specific Gibbs sampler with a Metropolis-Hastings step with a truncated Cauchy distribution as proposal. We have shown the application of our method with data from the Murcia population-based cancer registry, and we have compared our method with WinBUGS through two real examples and a simulation study.

In conclusion, the proposed algorithm converges to the posterior distribution faster than the methods implemented with WinBUGS. Furthermore the possibility of easily carrying out the estimations using our R functions is an important added value.

Appendix

For $j \in \{1, 2, \ldots, k\}$ and $\theta_{\sim j} = (\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_k) \in \mathbb{R}^k$ such that $\pi(\theta_{\sim j} \mid \mathbf{y}) = \int \pi(\theta \mid \mathbf{y}) d\theta_j > 0$, the full conditional distribution is $\pi(\theta_j \mid \mathbf{y}, \theta_{\sim j}) \propto \pi(\theta \mid \mathbf{y})$, $\theta_j \in \Theta_j$ where,

$$\Theta_j = \{\theta_j \in \mathbb{R}; \pi(\theta_j \mid \mathbf{y}, \theta_{\sim j}) > 0\}.$$

Proposition 1. If $\pi(\theta_{\sim j} \mid \mathbf{y}) > 0$ then the set Θ_j is the interval (a_j, b_j) where,

$$a_j = \max_{i \in A_j} \sum_{s \neq j} -z_{is}\theta_s / z_{ij}, \;\; A_j = \{i \in \mathbb{N}_n; z_{ij} < 0\},$$

and

$$b_j = \min_{i \in B_j} \sum_{s \neq j} -z_{is}\theta_s / z_{ij}, \;\; B_j = \{i \in \mathbb{N}_n; z_{ij} > 0\},$$

with the convention that $a_j = -\infty$ if $A_j = \emptyset$ and $b_j = +\infty$ if $B_j = \emptyset$.

Proof. Because of $\pi(\theta_{\sim j}|\mathbf{y}) > 0$, there exist $\theta_j^* \in \mathbb{R}$ such that $\pi(\theta_j^*, \theta_{\sim j}|\mathbf{y}) > 0$ and hence $\theta_j^* \in \Theta_j$. Since $\pi(\theta_j^*, \theta_{\sim j}|\mathbf{y}) > 0$ it follows that $z_{ij}\theta_j^* + \sum_{s \neq j} z_{is}\theta_s < 0$ for $i \in \mathbb{N}_n$ and then

$$\sum_{s \neq j} z_{is}\theta_s < 0, \, \forall i \in \mathbb{N}_n \;\; \text{such that}\, z_{ij} = 0.$$

Let θ_j be a real number. Then $\theta_j \in \Theta_j$ if and only if

$$z_{ij}\theta_j + \sum_{s \neq j} z_{is}\theta_s < 0, \, \forall i \in \mathbb{N}_n,$$

that is, if and only if

$$\theta_j > \sum_{s \neq j} -z_{is}\theta_s / z_{ij}, \, \forall i \in A_j,$$

$$\theta_j < \sum_{s \neq j} -z_{is}\theta_s / z_{ij}, \, \forall i \in B_j$$

and

$$\sum_{s \neq j} z_{is}\theta_s < 0, \, \forall i \in \mathbb{N}_n \;\; \text{such that}\, z_{ij} = 0.$$

It follows that $\Theta_j = (a_j, b_j)$.

R functions

The proposed Metropolis-within-Gibbs algorithm has been implemented in R using the following functions.

```
gibbsLogBinomial=function(j){
ztheta=Z[,-j]%*%matrix(theta[-j],ncol=1)
A=Aind[[j]];B=Bind[[j]]
suma1=sum(Z[,j]<0);a=-Inf
if(suma1!=0){a=max(-ztheta[A]/Z[A,j])}
suma2=sum(Z[,j]>0);b=Inf
if(suma2!=0){b=min(-ztheta[B]/Z[B,j])}
```

```
u=runif(1,0,1);location=theta.hat[j]
thetaj.star=location-tan((u-1)*atan(a-location)+u*atan(location-b))
theta.new=theta;theta.new[j]=thetaj.star
p.new=exp(Z[,j]*(thetaj.star-theta[j]))*p
logvalue.new=sum(log(p.new[y==1]))+sum(log(1-p.new[y==0]))
priortheta.new=prior(theta.new)

rho=exp(logvalue.new-logvalue);rho=rho*priortheta.new/priortheta
rho=rho*(1+(thetaj.star-location)^2)/(1+(theta[j]-location)^2)
rho=min(1,rho)

logvalue<<-logvalue;theta<<-theta;p<<-p
priortheta<<-priortheta
u=runif(1,0,1)
if(u<rho){theta<<-theta.new;logvalue<<-logvalue.new;p<<-p.new;
priortheta<<-priortheta.new}
}

prior=function(theta){return(1)}
inicial.beta=function(){
coef=summary(glm(y ~ 1,family=binomial))$coeff
mu=coef[1,1];serror=coef[1,2]
musim=rnorm(1,mu,serror)
beta1=log(exp(musim)/(1+exp(musim)))
return(c(beta1,rep(0,k-1)))}

initialize=function(){
#Reparameterization
X<<-model.ini$x;n<<-nrow(X);beta=model.ini$coeff
Sigma<<-summary(model.ini)$cov.unscaled
L<<-chol(Sigma);Z<<-X%*%t(L)
model.ini.0<<-glm(y ~ Z-1,family=poisson,x=TRUE)
theta.hat<<-solve(t(L))%*%beta;k<<-ncol(Z)

Aind<<-{}
for(j in 1:k){Aind[[j]]<<-(1:n)[Z[,j]<0]}
Bind<<-{}
for(j in 1:k){Bind[[j]]<<-(1:n)[Z[,j]>0]}

#Initial point.
punto<<-solve(t(L))%*%inicial.beta()
theta<<-punto;p<<-exp(Z%*%theta)
logvalue<<-sum(log(p[y==1]))+sum(log(1-p[y==0]))
priortheta<<-prior(theta)
```

```
}
```

Using the R function gibbsLogBinomial with the Breast cancer mortality example

```
#The data
datos<-rbind(cbind(rep(1,12),rep(1,12),c(rep(1,2),rep(0,10))),
cbind(rep(1,55),rep(2,55),c(rep(1,5),rep(0,50))),
cbind(rep(2,22),rep(1,22),c(rep(1,9),rep(0,13))),
cbind(rep(2,74),rep(2,74),c(rep(1,17),rep(0,57))),
cbind(rep(3,14),rep(1,14),c(rep(1,12),rep(0,2))),
cbind(rep(3,15),rep(2,15),c(rep(1,9),rep(0,6))))
datos<-data.frame(datos)
names(datos)<-c("Stage","Receptor_Level","Dead")
#Recoding Receptor_level
datos$Receptor_Level=as.integer(datos$Receptor_Level==1)
#Outcome
y=datos$Dead
##############################################################
################ Runing the MCMC algorithm ###################
#Poisson model. The following line depends on covariates
model.ini=glm(y˜factor(Receptor_Level)+factor(Stage),
family=poisson,data=datos,x=TRUE)
#The following lines compute the need input for
#the algorithm and fix the lengtht of the chain to 10000
initialize()
longChain=10000
theta.sim=matrix(rep(NA,longChain*k),ncol=k)
#Finally the chain is simulated as follows
for(h in 1:longChain){theta.sim[h,]=theta
   for(j in 1:k){gibbsLogBinomial(j)}}
beta.sim=theta.sim%*%L
#The object beta.sim containts the simulations
#Posterior estimation of exp(beta) using the coda package
library(coda)
RR=mcmc(exp(beta.sim))
summary(RR)
```

Acknowledgements

The authors thank the reviewers, the Editor and the Associate Editor for their constructive comments.

References

1. McNutt LA, Wu C, Xue X, Haffner JP. Estimating the relative risk in cohort studies and clinical trials of common outcomes. *American Journal of Epidemiology* 2003; **157**:940–943. DOI: 10.1093/aje/kwg074
2. Deddens JA, Petersen MR, Lei X. Estimation of prevalence ratios when PROC GENMOD does not converge. *Paper 270-28. Proceedings of the 28th Annual SAS Users Group International Conference, Seattle, Washington, March 30-April 2, 2003*. Downloaded from http://www2.sas.com/proceedings/sugi28/270-28.pdf on the 3rd March 2015.
3. Greenland S. Model-based estimation of relative risks and other epidemiologic measures in studies of common outcomes and in case-control studies. *American Journal of Epidemiology* 2004; **160**:301–305. DOI: 10.1093/aje/kwh221
4. Spiegelman D, Hertzmark E. Easy SAS calculations for risk or prevalence ratios and differences. *American Journal of Epidemiology* 2005; **162**:199–200. DOI: 10.1093/aje/kwi188
5. Petersen MR, Deddens JA. RE: ”Easy SAS calculations for risk or prevalence ratios and differences”. *American Journal of Epidemiology* 2006; **163**:1158–1159. DOI: 10.1093/aje/kwj162
6. Deddens JA, Petersen MR. Approaches for estimating prevalence ratios. *Occupational and Environmental Medicine* 2008; **65**:501-506. DOI: 10.1136/oem.2007.034777
7. Zou GY. A modified Poisson regression approach to prospective studies with binary data. *American Journal of Epidemiology* 2004; **159**:702–706.
8. Savu A, Liu Q, Yasui Y. Estimation of relative risk and prevalence ratio. *Statistics in Medicine* 2010; **29**:2269–2281. DOI: 10.1002/sim.3989
9. Chu H, Cole SR. Estimation of risk ratios in cohort studies with common outcomes: a Bayesian approach. *Epidemiology* 2010; **21**:855-862. DOI: 10.1097/EDE.0b013e3181f2012b
10. Spiegelhalter DJ, Thomas A, Best NG. WinBUGS User Manual, Version 1.4. Cambridge, United Kingdom: Medical Research Council Biostatistics Unit; 2003.
11. Hamra G, MacLehose R, Richardson D. Markov chain Monte Carlo: an introduction for epidemiologists. *International Journal of Epidemiology* 2013; **42**:627–634. DOI: 10.1093/ije/dyt043
12. R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.
13. Smith AFM, Roberts GO. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 1993; **55**:3-23.
14. Minicozzi P, Caldarella A, Giacomin A, Ponz de Leon M, Cesaraccio R, Falcini F, Fusco M, Iachetta F, Pellegri C, Tumino R, Capocaccia R, Sant M. Looking at differences in stage and treatment of colorectal cancers across Italy: a EUROCARE-5 high resolution study. *Tumori* 2012; **98**: 671-677. DOI: 10.1700/1217.13488
15. Allemani C, Rachet B, Weir HK, Richardson LC, Lepage C, Faivre J, Gatta G, Capocaccia R, Sant M, Baili P, Lombardo C, Aareleid T, Ardanaz E, Bielska-Lasota M, Bolick S, Cress R, Elferink M, Fulton JP, Galceran J, Gzdz S, Hakulinen T, Primic-Zakelj M, Rachtan J, Diba CS, Snchez MJ, Schymura MJ, Shen T, Tagliabue G, Tumino R, Vercelli M, Wolf HJ, Wu XC, Coleman MP. Colorectal cancer survival in the USA and Europe: a CONCORD high-resolution study. *BMJ Open* 2013; **3**:e003055. DOI: 10.1136/bmjopen-2013-003055
16. Plummer M, Best N, Cowles K, Vines K. CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News* 2006; **6**:7–11.
17. Hosmer DW, Lemeshow S. *Applied Logistic Regression*. 2nd ed. John Wiley and Sons: New York, 2000.

Table 1. Colon cancer example. Probability of being alive five years after diagnosis, having survived one year. Numbers in this table are the maximum likelihood estimates and the confidence intervals obtained using logistic regression, and posterior means and credible intervals obtained with our proposed algorithm and with WinBUGS for log-binomial regression models.

Variable (N)	Logistic OR (CI 95%)	Proposed method RR (CI 95%)	WinBUGS RR (CI 95%)
Sex			
Male (1,622)	1	1	1
Female (1,327)	1.38 (1.17,1.63)	1.09 (1.05,1.14)	1.09 (1.05,1.14)
Stage			
I (334)	9.95 (6.22,15.90)	2.46 (1.93,3.18)	2.72 (2.28,3.13)
II (1,177)	7.32 (4.91,10.91)	2.36 (1.87,3.04)	2.60 (2.20,3.00)
III (891)	2.51 (1.69,3.73)	1.70 (1.33,2.19)	1.87 (1.57,2.18)
IV (131)	1	1	1
Unknown (416)	5.12 (3.33,7.88)	2.18 (1.71,2.81)	2.40 (2.01,2.80)
Age			
55-59 (318)	2.73 (1.99,3.74)	1.28 (1.19,1.38)	1.28 (1.19,1.38)
60-64 (478)	2.14 (1.65,2.79)	1.24 (1.15,1.33)	1.24 (1.15,1.33)
65-69 (664)	1.67 (1.32,2.11)	1.19 (1.10,1.28)	1.19 (1.10,1.27)
70-74 (789)	1.65 (1.32,2.07)	1.16 (1.08,1.25)	1.16 (1.08,1.25)
75-79 (700)	1	1	1

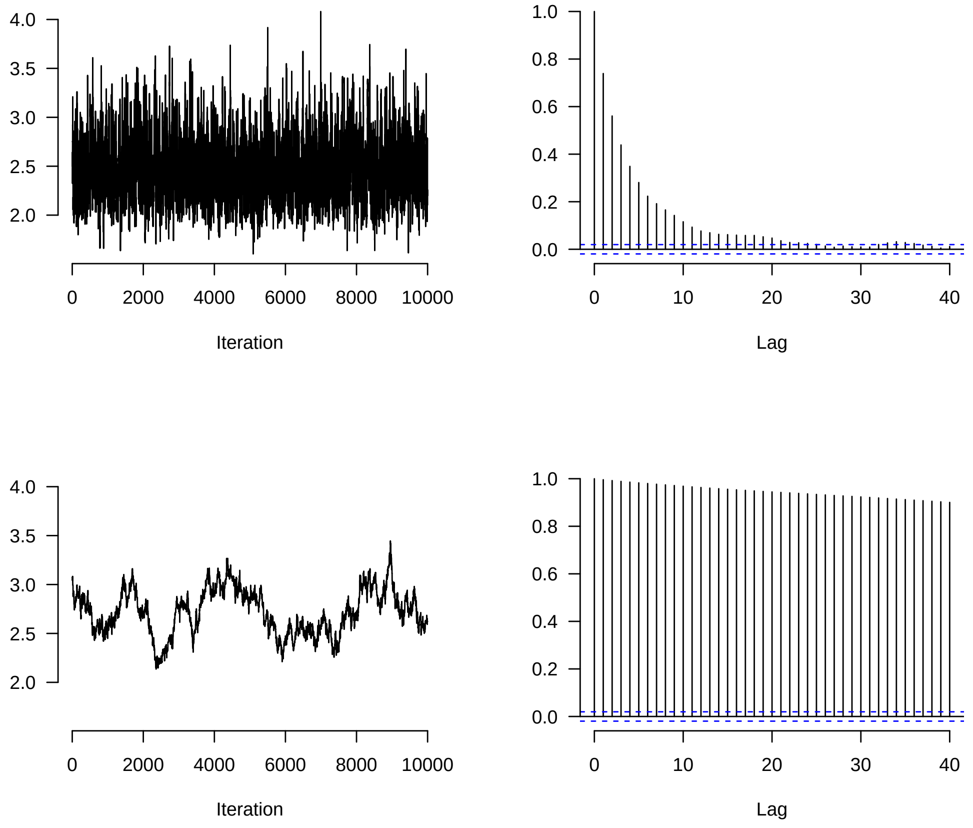

Figure 1. Colon cancer example. Trace plots and autocorrelation functions obtained with the proposed algorithm (first row) and with WinBUGS (second row). Results for the parameter $\exp(\beta_3)$.

Table 2. Posterior mean, posterior standard deviation and the posterior quantiles $Q(0.025)$ and $Q(0.975)$ of $\exp(\beta_j)$, $j = 1, 2, 3, 4$, using a chain of length 1,000,000 obtained with our method for the breast cancer mortality example.

	Posterior mean	Posterior standard deviation	$Q(0.025)$	$Q(0.975)$
$\exp(\beta_1)$	0.093	0.034	0.039	0.169
$\exp(\beta_2)$	1.578	0.332	1.051	2.346
$\exp(\beta_3)$	2.920	1.347	1.279	6.321
$\exp(\beta_4)$	6.560	2.988	2.925	14.106

Table 3. Effective sample size (ESS) for the breast cancer mortality example. Means and standard deviation of the 500 effective sample sizes obtained with WinBUGS and the proposed method, based on Markov chains of length 9,500.

	WinBUGS	Proposed method
	Mean (SD)	Mean (SD)
$\exp(\beta_1)$	71.9 (13.3)	5,192.1 (218.7)
$\exp(\beta_2)$	324.8 (30.6)	4,047.4 (209.6)
$\exp(\beta_3)$	60.2 (17.8)	5,166.7 (275.3)
$\exp(\beta_4)$	59.2 (16.5)	4,486.8 (259.6)

Table 4. Posterior mean, posterior standard deviation and posterior quantiles $Q(0.025)$ and $Q(0.975)$ of $\exp(\beta_j)$, $j = 1, 2, 3, 4$, using a chain of length 1,000,000 obtained with our method for the low birth weight example.

	Posterior mean	Posterior standard deviation	$Q(0.025)$	$Q(0.975)$
$\exp(\beta_1)$	0.161	0.053	0.078	0.284
$\exp(\beta_2)$	1.240	0.274	0.779	1.859
$\exp(\beta_3)$	1.584	0.337	1.029	2.348
$\exp(\beta_4)$	1.748	0.489	0.934	2.846
$\exp(\beta_5)$	1.567	0.369	0.973	2.415
$\exp(\beta_6)$	1.123	0.360	0.558	1.953
$\exp(\beta_7)$	1.227	0.322	0.730	1.985
$\exp(\beta_8)$	0.935	0.281	0.480	1.593
$\exp(\beta_9)$	0.529	0.286	0.116	1.197
$\exp(\beta_{10})$	1.729	0.360	1.120	2.528

Table 5. Effective sample size (ESS) for the low birth weight example. Mean and standard deviation of the 500 effective sample sizes obtained with WinBUGS and the proposed method, based on Markov chains of length 9,500.

	WinBUGS	Proposed method
	Mean (SD)	Mean (SD)
$\exp(\beta_1)$	83.7 (12.8)	4,416.3 (237.7)
$\exp(\beta_2)$	432.6 (43.8)	4,325.7 (241.6)
$\exp(\beta_3)$	240.1 (39.2)	3,842.3 (194.3)
$\exp(\beta_4)$	309.8 (65.5)	4,148.8 (215.4)
$\exp(\beta_5)$	245.2 (43.0)	4,093.9 (231.0)
$\exp(\beta_6)$	264.0 (68.4)	4,928.2 (283.3)
$\exp(\beta_7)$	191.1 (48.9)	4,817.9 (286.0)
$\exp(\beta_8)$	431.2 (105.1)	5,621.1 (315.3)
$\exp(\beta_9)$	874.0 (227.8)	5,443.4 (275.0)
$\exp(\beta_{10})$	315.5 (32.3)	2,438.6 (150.1)

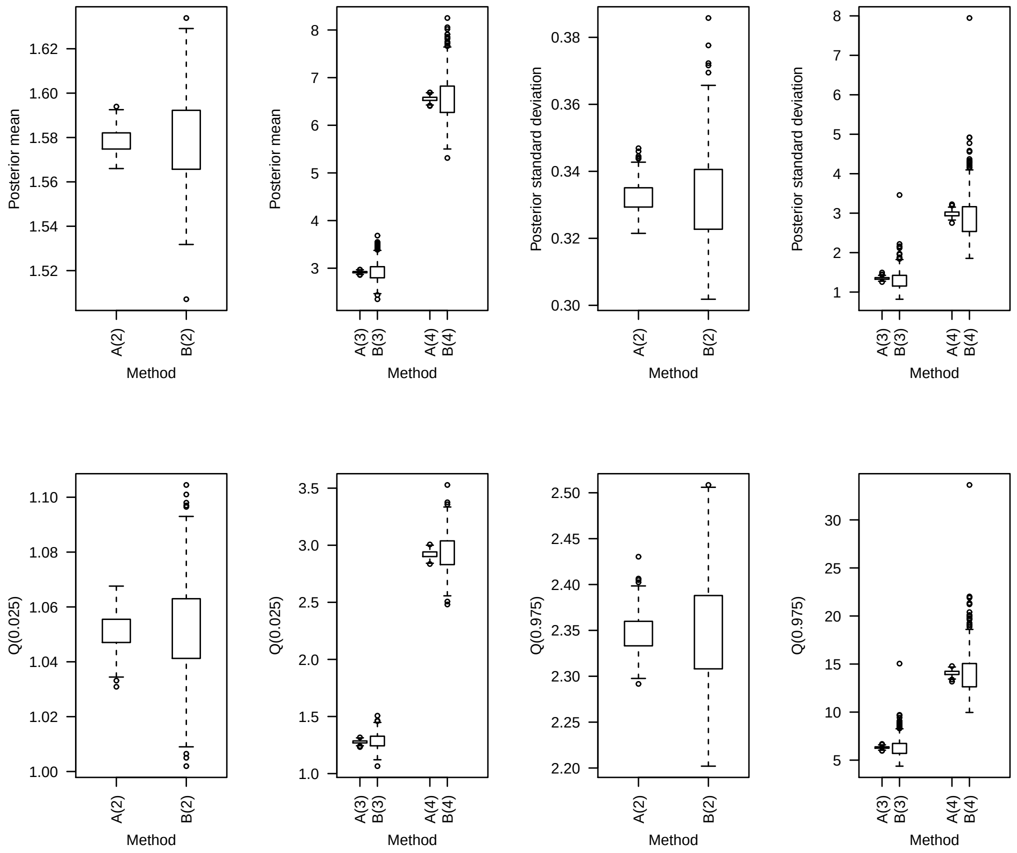

Figure 2. Boxplots of the approximations obtained with the proposed method and with WinBUGS for the breast cancer mortality example. Each boxplot is based on 500 approximations to the posterior mean, the posterior standard deviation and the quantiles Q(0.025) and Q(0.975) of the posterior distribution of $\exp(\beta_i)$, $i = 2, 3, 4$. Each approximation is based on a Markov chain of length 10,000. A(i) denotes the boxplot corresponding to the approximations obtained with the proposed method for the parameter $\exp(\beta_i)$. B(i) denotes the boxplot corresponding to the approximations obtained with WinBUGS for the parameter $\exp(\beta_i)$.

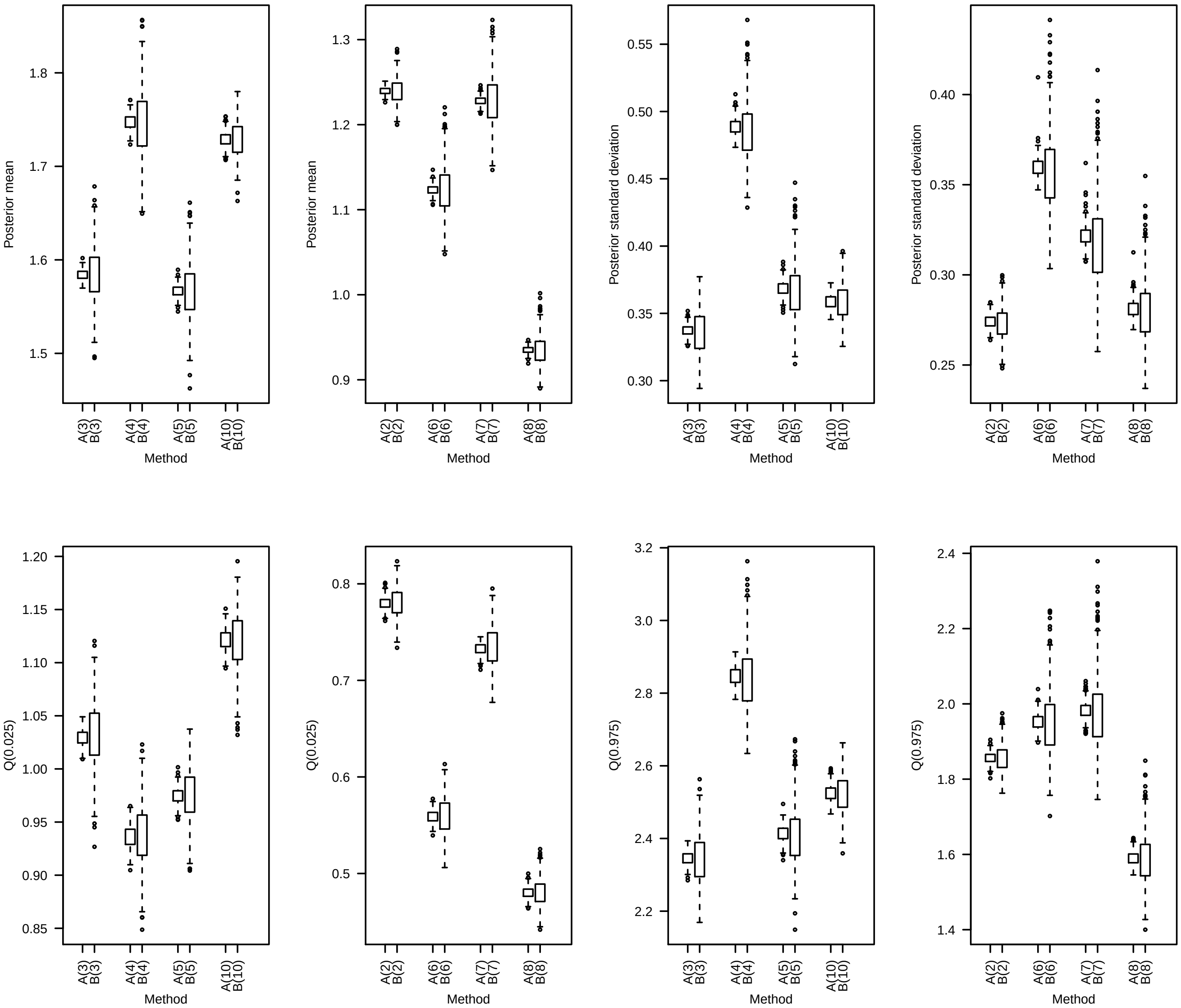

Figure 3. Boxplots of the approximations obtained with the proposed method and with WinBUGS for the low birth weight example. Each boxplot is based on 500 approximations to the posterior mean, the posterior standard deviation, and the quantiles Q(0.025) and Q(0.975) of the posterior distribution of $\exp(\beta_i)$, $i = 2, 3, 4, 5, 6, 7, 8, 10$. Each approximation is based on a Markov chain of length 10,000. A(i) denotes the boxplot corresponding to the approximations obtained with the proposed method for the parameter $\exp(\beta_i)$. B(i) denotes the boxplot corresponding to the approximations obtained with WinBUGS for the parameter $\exp(\beta_i)$.

Table 6. Efficiency and coverage probability for the parameters $RR_i = \exp(\beta_i),\ i = 1, 2, 3$, in the simulation study. Mean and standard deviation (sd) of the 1,000 values of Eff for each set of true values of the parameters $(\exp(\beta_1), \exp(\beta_2), \exp(\beta_3))$, coverage (percentage of times that the approximate 95% credible interval covers the true value of the parameter), and average length of the 95% posterior credible intervals.

RR_1						
Set of	WinBUGS			The Proposal		
parameters	Mean(sd)	coverage	length	Mean(sd)	coverage	length
(0.40,1.0,1.0)	7.1(1.2)	92.1	0.3	803.3(50.0)	94.5	0.3
(0.35,1.5,1.0)	6.5(1.2)	93.3	0.3	791.9(55.1)	94.9	0.3
(0.30,2.0,1.0)	6.1(1.2)	94.9	0.2	772.8(63.5)	95.8	0.3
(0.35,1.0,1.5)	6.6(1.2)	94.1	0.3	795.3(61.1)	94.1	0.3
(0.30,1.5,1.5)	6.5(1.1)	93.4	0.2	773.2(82.8)	94.2	0.2
(0.25,2.0,1.5)	6.3(1.1)	94.2	0.2	739.5(102.4)	95.1	0.2
(0.30,1.0,2.0)	6.2(1.1)	93.5	0.2	774.1(70.5)	94.7	0.3
(0.25,1.5,2.0)	6.4(1.1)	93.2	0.2	750.0(100.0)	94.3	0.2
(0.20,2.0,2.0)	6.6(1.2)	94.3	0.2	719.5(120.0)	95.8	0.2
(0.25,1.0,3.0)	5.6(1.1)	92.9	0.2	702.0(107.8)	95.8	0.2
(0.20,1.5,3.0)	6.3(1.2)	96.1	0.2	628.6(166.6)	96.1	0.2
(0.15,2.0,3.0)	6.5(1.2)	95.7	0.1	658.6(153.0)	95.7	0.1
RR_2						
Set of	WinBUGS			The Proposal		
parameters	Mean(sd)	coverage	length	Mean(sd)	coverage	length
(0.40,1.0,1.0)	11.5(3.0)	93.1	1.0	796.6(54.3)	95.5	1.0
(0.35,1.5,1.0)	9.8(2.6)	93.6	1.4	767.9(59.7)	94.8	1.4
(0.30,2.0,1.0)	8.8(2.5)	93.9	1.8	742.6(70.0)	95.8	1.9
(0.35,1.0,1.5)	12.4(3.5)	93.9	0.9	812.2(59.1)	94.6	0.9
(0.30,1.5,1.5)	11.2(3.1)	93.8	1.3	784.4(76.6)	94.9	1.3
(0.25,2.0,1.5)	10.1(2.8)	93.8	1.7	741.2(101.2)	95.3	1.7
(0.30,1.0,2.0)	13.1(4.0)	93.0	0.9	821.8(61.0)	95.6	0.9
(0.25,1.5,2.0)	12.4(3.6)	94.0	1.2	793.7(84.2)	96.6	1.3
(0.20,2.0,2.0)	11.6(3.0)	94.6	1.8	750.7(110.8)	96.1	1.8
(0.25,1.0,3.0)	14.8(4.6)	94.8	0.8	826.4(87.0)	96.6	0.8
(0.20,1.5,3.0)	14.3(3.9)	94.7	1.1	754.5(141.8)	96.0	1.1
(0.15,2.0,3.0)	12.6(3.3)	95.1	1.7	737.5(137.6)	96.5	1.8
RR_3						
Set of	WinBUGS			The Proposal		
parameters	Mean(sd)	coverage	length	Mean(sd)	coverage	length
(0.40,1.0,1.0)	11.9(3.7)	92.3	1.0	780.9(54.1)	95.4	1.0
(0.35,1.5,1.0)	13.1(4.0)	94.2	0.9	780.8(59.6)	94.8	0.9
(0.30,2.0,1.0)	13.9(4.3)	92.3	0.9	775.5(63.8)	94.8	0.9
(0.35,1.0,1.5)	9.6(2.7)	92.7	1.4	781.6(61.3)	96.0	1.4
(0.30,1.5,1.5)	11.4(3.3)	93.4	1.2	772.8(73.0)	95.3	1.3
(0.25,2.0,1.5)	12.5(3.5)	93.3	1.2	761.1(83.1)	94.3	1.3
(0.30,1.0,2.0)	8.6(2.5)	94.0	1.8	763.6(67.6)	94.7	1.8
(0.25,1.5,2.0)	10.5(3.0)	93.9	1.6	758.0(87.8)	93.7	1.7
(0.20,2.0,2.0)	12.4(3.6)	94.4	1.7	747.5(104.5)	95.5	1.8
(0.25,1.0,3.0)	7.3(2.1)	94.2	2.4	702.4(102.7)	96.0	2.5
(0.20,1.5,3.0)	9.1(2.5)	95.7	2.3	651.0(151.3)	95.6	2.4
(0.15,2.0,3.0)	10.6(3.0)	93.9	2.6	687.2(141.0)	95.8	2.7