Title:

MC(MC)MC: Exploring Monte Carlo integration within MCMC for Mark-Recapture Models with Individual Covariates

Author Information:

Simon Bonner[1,2] and Matthew Schofield[1]

1 Department of Statistics
University of Kentucky
Rm. 311 Multidisciplinary Science Building
725 Rose Street
Lexington, KY 40536-0082
USA

2 Email: simon.bonner@uky.edu
Phone: 859-257-4950
Fax: 859-323-1973

Running Head:

Modeling Mark-Recapture Data via MCWM

Word Count:

This manuscript contains 6709 words including figure and table captions and headers, but excluding the title page and equations.

Keywords

Data Augmentation; Mark-recapture; Markov Chain Monte Carlo; MCWM; Monte Carlo within MCMC; Monte Carlo Integration

Summary

1. Estimating abundance from mark-recapture data is challenging when capture probabilities vary among individuals.

2. Initial solutions to this problem were based on fitting conditional likelihoods and estimating abundance as a derived parameter. More recently, Bayesian methods using full likelihoods have been implemented via reversible jump Markov chain Monte Carlo sampling (RJMCMC) or data augmentation (DA). The latter approach is easily implemented in available software and has been applied to fit models that allow for heterogeneity in both open and closed populations. However, both RJMCMC and DA may be inefficient when modeling large populations.

3. We describe an alternative approach using Monte Carlo (MC) integration to approximate the posterior density within a Markov chain Monte Carlo sampling scheme. We show how this Monte Carlo within MCMC (MCWM) approach may be used to fit a simple, closed population model including a single individual covariate, and present results from a simulation study comparing RJMCMC, DA, and MCWM. We found that MCWM can provide accurate inference about population size and can be more efficient than both RJMCMC and DA. The efficiency of MCWM can also be improved by using advanced MC methods like antithetic sampling.

4. Finally, we apply MCWM to estimate the abundance of meadow voles (*Microtus pennsylvanicus*) at the Patuxent Wildlife Research Center in 1982 allowing for capture probabilities to vary as a function body mass.

1 Introduction

Individual variation is a key driver of evolution and an important consideration in modeling the demographics of many populations. However, individual heterogeneity presents a challenge in the analysis of mark-recapture data – particularly when the goal is to estimate abundance. In practice, differences in the behavior of individuals in a population may be modeled as functions of individual covariates or random effects. In either case, the likelihood function will include integrals to account for all possible values of the unobserved effects. These integrals may be difficult to compute if multiple covariates/random effects are included or if a single individual covariate/random effect changes over time, which makes evaluating the true likelihood for the entire population problematic.

Intractable likelihoods pose a general problem in statistics, and several solutions have been proposed within the Bayesian framework. We explore Monte Carlo integration within Markov chain Monte Carlo sampling (MCWM) to obtain inference from mark-recapture models with individual heterogeneity. While we focus on modeling the effects of individual covariates, the same methods can be applied to models including random effects or a combination of the two.

One way to avoid the problem with intractable likelihoods is to estimate abundance with a conditional likelihood approach. Huggins (1989) and Alho (1990) presented methods for estimating the size of a closed population when the capture probability depends on an individual covariate. Likelihoods which condition on at least one capture are fit to the data from the marked individuals and used to estimate capture probability as a function of the covariate. Abundance is then estimated using a Horvitz-Thompson estimator. These methods were later extended to open population models by McDonald and Amstrup (2001). However, these models are restrictive and can only be used if the covariate is completely observed for the marked individuals (i.e., the covariate is constant or changes deterministically like age).

Alternatively, Bayesian inference via Markov chain Monte Carlo (MCMC) has been applied to fit models allowing for the effects of time-varying, individual covariates or other covariates that are only partially observed for the marked individuals. Dupuis (1995) applied Bayesian methods to model the effects of discrete covariates on survival of individuals in an open population (i.e., the multi-state model). Following this, Pollock (2002) suggested that a Bayesian approach could be applied for the particular case of continuous, time-varying, individual covariates and noted that: “Bayesian methods automatically integrate out unobserved random variables using numerical integration or Markov Chain Monte-Carlo sampling methods” (Pollock, 2002, pg. 97). Bonner and Schwarz (2006) applied Bayesian inference via MCMC to model the effects of time dependent covariates on individual capture and survival probabilities in the Cormack-Jolly-Seber (CJS) model. King et al. (2006) described a similar approach and provided methods of variable selection while Gimenez et al. (2006) incorporated semi-parametric regression to allow for non-linear effects of the covariate. Royle et al. (2007) and Royle (2009) later developed MCMC based methods to make inference about the size of a closed population when capture probabilities vary among individuals. Their method is based on augmenting the observed data with a large number of zero capture histories representing a pool of individuals that may have been alive but never captured and has become known as the data augmentation (DA) approach. This method is appealing because it provides a conceptually simple framework that can be applied to many models and is easily implemented in the BUGS language. More recently, Schofield and Barker (2011) and Royle and Dorazio (2012) have shown how the same methods may be applied to model open populations with individual heterogeneity. Alternatively, Bayesian inference regarding the size of an open or closed population with individual heterogeneity may implemented with the reversible jump MCMC (RJMCMC) algorithm as described by King and Brooks (2008).

Our current work is motivated by our experiences applying DA and RJMCMC to a variety of mark-recapture data sets. Both DA and RJMCMC avoid the need for explicit integration by working with complete data likelihoods (CDL) in place of the observed data likelihood. These CDL are constructed by adding extra, unobserved random variables to the data which would simplify computation of the likelihood, if observed (see e.g. Dempster et al. 1977 and Gelman et al. 2003, Section7.2).

We have found that the chains constructed by these algorithms may be computationally inefficient[1] in that they mix poorly and take a long time to generate a representative sample from the posterior distribution. This seems especially true when the models include time-dependent, individual covariates or other multidimensional covariates which make the likelihood difficult to evaluate numerically. All MCMC methods work by constructing a Markov chain which has the posterior distribution as its unique stationary distribution. Samples from the posterior distribution are generated by simulating sufficiently long realizations of the Markov chain, and these samples are used to estimate posterior

[1]We use efficiency to refer to computational efficiency of the different sampling algorithms not statistical efficiency. One algorithm is more efficient than another if it requires less time to provide the same amount of information about the posterior distribution.

summary statistics. The challenge with DA and RJMCMC is that a lot of time may be spent updating the extra variables added to the CDL when a small fraction of the population is captured and marked. Moreover, we have found that the chains can have high autocorrelation meaning that large samples are needed to estimate posterior summary statistics accurately.

We explore the use of MCWM as an alternative to these algorithms for fitting mark-recapture models with individual covariates. We focus on a simple, closed population model with one individual covariate as an example of the method and provide results of a simulation study comparing MCWM, DA, and RJMCMC. We also apply our method to data on meadow voles (*Microtus pennsylvanicus*) collected at the Patuxent Wildlife Research Center in 1981 and 1982 (Nichols et al., 1992) and compare the results with DA and RJMCMC. Although this data was collected using a robust design, we only consider the information from the final primary period and model capture probability as a function of a vole's average observed body mass. Previous analysis of this data has shown a significant, positive relationship between capture probability and body mass (Schofield and Barker, 2011), and abundance estimates which ignore this heterogeneity would be biased.

2 Methods

We describe MCWM and compare it with the alternative RJMCMC and DA algorithms for the following simple model. Suppose that the population of interest is closed and that the capture probability for each individual is a linear function of a normally distributed covariate on the logit scale. Assuming no behavioral effects, time effects, or losses on capture, the number of times the i^{th} individual is captured on T occasions, Y_i, can be modeled as:

$$Y_i|p_i \overset{ind}{\sim} \text{Binomial}(T, p_i), \quad i = 1, \ldots, N$$

where N is abundance and

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_i \text{ and } x_i \overset{iid}{\sim} N(\mu, \sigma^2).$$

Further, suppose that $\beta_0 = 0$, $\beta_1 = 1$, and $\sigma^2 = 1$ so that the only unknown parameters are μ and N. Let n denote the number of individuals captured at least one time and let $\boldsymbol{Y}^{obs} = (y_1, \ldots, y_n)'$ and $\boldsymbol{X}^{obs} = (x_1, \ldots, x_n)'$ represent the observed data. The observed data likelihood is:

$$L(\mu, N|\boldsymbol{Y}^{obs}, \boldsymbol{X}^{obs}) = \binom{N}{n} P_0(\mu)^{N-n} \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{T-y_i}\phi(x_i - \mu)$$

where $\phi(z)$ represents the standard normal density function and $P_0(\mu)$ is the probability that a randomly selected individual is never captured. That is:

$$P_0(\mu) = \int_{-\infty}^{\infty} \left(\frac{1}{1+\exp(x)}\right)^T \phi(x-\mu)\ dx. \tag{1}$$

To complete the Bayesian specification we define prior distributions for the two unknown parameters. We assume independent priors for μ and N such that the posterior density satisfies:

$$\pi(\mu, N|\boldsymbol{Y}^{obs}, \boldsymbol{X}^{obs}) \propto L(\mu, N|\boldsymbol{Y}^{obs}, \boldsymbol{X}^{obs})\pi(\mu)\pi(N).$$

Specifically, we have selected a conjugate normal prior for μ, $\mu \sim N(0, \tau^2)$ with τ^2 fixed, and the Jeffrey's prior for N, $\pi(N) \propto N^{-1}$, as recommended by Link (2013).

The posterior density is not tractable even for this simple model and so it is necessary to sample from the posterior distribution to make inference about μ and N. Supposing that it was in fact possible to evaluate $P_0(\mu)$ directly, the likelihood in eqn. (1) could be computed explicitly and values from the posterior distribution could be generated by a standard MCMC implementation. The full conditional distribution of N would follow a negative binomial distribution so that values of N could be generated directly (a so-called Gibbs sampling step)[2]. The full conditional distribution of μ would not be tractable, but values of μ could be generated from a slightly more complicated Metropolis-Hastings step. This involves proposing a new value for μ from some distribution conditional on the current value, denoted by $q(\cdot|\mu)$, and accepting or rejecting this proposal according to the Hastings ratio (see for example Gilks et al. (1996, pg. 5–8)). Explicitly, let $\mu^{(t)}$ and $N^{(t)}$ represent the values of μ and N generated on the t^{th} iteration. The next values would be generated in two steps by:

A) Updating $\mu^{(t)}$ given $N^{(t)}$, $\boldsymbol{X}^{obs}$, and $\boldsymbol{Y}^{obs}$ via a MH step:

1) Propose $\mu' \sim q(\mu|\mu^{(t)})$

2) Accept μ' and set $\mu^{(t+1)} = \mu'$ with probability $\alpha = \min(1, r(\mu^{(t)}, \mu'))$ where:

$$r(\mu^{(t)}, \mu') = \frac{\pi(\mu'|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})}{\pi(\mu^{(t)}|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})} \cdot \frac{q(\mu^{(t)}|\mu')}{q(\mu'|\mu^{(t)})}.$$

[2] The negative binomial may be considered as a distribution on either the number of trials or number of failures until a specified number of successes occurs. We consider the distribution of the number of trials until n successes are reached so that $N \geq n$.

Otherwise, set $\mu^{(t+1)} = \mu^{(t)}$.

B) Updating $N^{(t)}$ given $\mu^{(t+1)}$, $\boldsymbol{X}^{obs}$, and $\boldsymbol{Y}^{obs}$ via a Gibb's sampling step:

$$N^{(t+1)} \sim \text{Neg. Bin.}(n, 1 - P_0(\mu^{(t+1)})).$$

Under general conditions on $q(\cdot|\mu)$, the distribution of $(\mu^{(t)}, N^{(t)})$ would converge to the posterior distribution as $t \to \infty$. If t were big enough then $(\mu^{(t)}, N^{(t)}), \ldots, (\mu^{(t+s)}, N^{(t+s)})$ could be considered as approximate (in some cases, exact) draws from the posterior distribution and used to estimate posterior summary statistics (see Gilks et al. (1996) for further details). Of course, this algorithm cannot be implemented because $P_0(\mu)$ cannot be computed.

2.1 Complete Data Likelihoods

Both RJMCMC and DA avoid the need to compute $P_0(\mu)$ directly by constructing posterior distributions from CDLs that do not include the integral in eqn. (1). As mentioned above, these CDLs are formed by expanding the model to include additional, unobserved data that simplify the likelihood.

The CDL for RJMCMC is constructed by modeling the hypothetical data for all N individuals in the population. For the simple model, the additional random variables comprise the covariates for the $N - n$ unobserved individuals denoted by

$\boldsymbol{X}_N^{miss} = (x_{n+1}, \ldots, x_N)'$. The CDL for RJMCMC is:

$$L_{RJ}(\mu, N, \boldsymbol{X}_N^{miss} | \boldsymbol{Y}^{obs}, \boldsymbol{X}^{obs}) = \binom{N}{n} \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{T-y_i}\phi(x_i - \mu)$$
$$\prod_{i=n+1}^{N} \left(\frac{1}{1+\exp(x_i)}\right)^T \phi(x_i - \mu).$$

The posterior distribution is constructed by assigning priors to the parameters μ and N, exactly as above. Summary statistics including posterior means, standard deviations, and credible intervals are then approximated by sampling values from the joint posterior distribution of μ, N and $\boldsymbol{X}_N^{miss}$.

The full conditional distribution of N for RJMCMC does not have a simple form and cannot be updated by Gibbs sampling. In fact, the update of N requires a reversible jump (RJ) step that is more complicated than the standard MH update because the dimension of $\boldsymbol{X}_N^{miss}$ depends on N. In the RJ step, a new value for N is proposed as in an MH step but a corresponding proposal for $\boldsymbol{X}_N^{miss}$ must also be constructed by adding or deleting elements to obtain the correct number of covariates. The proposals for N and $\boldsymbol{X}_N^{miss}$ are then accepted or rejected as a single unit. Further to this, the elements of $\boldsymbol{X}_N^{miss}$ must be updated separately outside of the reversible jump step. The full conditionals for these values are not tractable, and these values must be updated through $N - n$ separate MH steps (see Schofield and Barker, 2011, for details).

As an alternative, the DA algorithm of Royle et al. (2007) constructs a CDL by modeling the hypothetical data for a fixed super-population of size $M >> N$. The additional data for our simple model comprises the covariates for the $M - n$ unobserved individuals in the super-population, $\boldsymbol{X}_M^{miss} = (x_{n+1}, \ldots, x_M)'$, along with $M - n$ binary variables indicating which unobserved individuals are part of the realized population, denoted by $\boldsymbol{z} = (z_{n+1}, \ldots, z_M)'$. The CDL for DA is:

$$
\begin{aligned}
L_{DA}(\mu, \psi, \boldsymbol{z}, \boldsymbol{X}_M^{miss} | \boldsymbol{Y}^{obs}, \boldsymbol{X}^{obs}) = & \prod_{i=1}^{n} \psi p_i^{y_i} (1 - p_i)^{T - y_i} \phi(x_i - \mu) \\
& \prod_{i=n+1}^{M} \left[\psi \left(\frac{1}{1 + \exp(x_i)} \right)^T \right]^{z_i} (1 - \psi)^{1 - z_i} \phi(x_i - \mu).
\end{aligned}
$$

Here $\psi = P(z_i = 1)$ is the probability that an individual in the super-population is part of the realized population. The posterior distribution is constructed by assigning prior distributions to μ and ψ. We assign μ a conjugate normal prior, as above, and approximate the Jeffrey's prior for N by setting $\psi \sim \text{Beta}(.0001, 1)$, as described by Link (2013). Samples are then drawn from the posterior distribution of μ, ψ, and $\boldsymbol{z}$ with N treated as a derived quantity ($N = n + \sum_{i=n+1}^{M} z_i$).

In comparison with the RJMCMC algorithm, all of the updates in the DA algorithm may implemented with Gibbs or MH steps. However, the variables z_i and x_i must be updated for each unobserved individual on each iteration. These two values may be updated separately or together in a block MH step, but in either case the complexity of DA depends on M. We have found that RJMCMC and DA may both take a long time to run and the resulting chains may have high autocorrelation when N is large and the distribution of the covariate is complex.

2.2 Monte Carlo within MCMC

In short, MCWM is a generalization of the MH updater which uses Monte Carlo (MC) integration to approximate both the numerator and denominator of the Hastings ratio when the exact posterior density cannot be computed. For the simple example, this allows us to implement an approximation to the two step MCMC algorithm presented at the start of this section which avoids computations which depend on N or M as in RJMCMC and DA. We first show how MCWM can be applied to update μ for the simple model and then show that our solution also addresses the problem of updating N.

Consider the MH step for updating μ described on page 8. Given the current value, $\mu^{(t)}$, a proposal is generated from some distribution, $q(\mu|\mu^{(t)})$. This value is then accepted with probability $\alpha = \min(1, r(\mu^{(t)}, \mu'))$ where:

$$r(\mu^{(t)}, \mu') = \frac{\pi(\mu'|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})}{\pi(\mu^{(t)}|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})} \cdot \frac{q(\mu^{(t)}|\mu')}{q(\mu'|\mu^{(t)})}.$$

In MCWM, the Hastings ratio, $r(\mu^{(t)}, \mu')$, is replaced by an approximation:

$$\hat{r}(\mu^{(t)}, \mu') = \frac{\widehat{\pi}_K(\mu'|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})}{\widehat{\pi}_K(\mu^{(t)}|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})} \cdot \frac{q(\mu^{(t)}|\mu')}{q(\mu'|\mu^{(t)})}$$

where $\widehat{\pi}_K(\mu'|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$ and $\widehat{\pi}_K(\mu^{(t)}|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$ represent MC estimates of the full conditional density of $\mu^{(t)}$ and μ', as described below (A brief introduction to MC integration is also provided in the Supplementary Materials). Approximating the Hastings ratio in this way introduces extra variability into the MH algorithm, and the posterior distribution is no longer a stationary distribution of the chain. However, Theorem 9 of

Andrieu and Roberts (2009) shows that the stationary distribution of the chains generated by MCWM approximates the true posterior when the MC estimator is unbiased and the size of the MC sample, denoted by K, is large. In essence, if the algorithm is run for enough iterations and the MC samples are large enough then the MCWM updater will produce values that are approximately, but not exactly, distributed according to the full conditional, $\pi(\mu|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$.

The remaining challenge in implementing this algorithm is to develop an efficient MC estimator of $\pi(\mu|N, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$. The only term in $\pi(\mu|N, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$ which cannot be computed directly is $Q(\mu) = P_0(\mu)^{N-n}$, and so it is sufficient to develop an MC estimator for this value alone. An unbiased estimator of $Q(\mu)$ can be obtained by generating K sets of $N - n$ covariate values:

$$\tilde{x}_{ik} \overset{iid}{\sim} N(\mu, 1), \quad i = 1, \ldots, N-n; k = 1, \ldots, K$$

and then setting:

$$\widehat{Q(\mu)} = \frac{1}{K} \sum_{k=1}^{K} \left(\prod_{i=1}^{N-n} (1 - p(\tilde{x}_{ik}))^T \right).$$

However, this requires generating $N \times K$ random variables so that the complexity of this estimator depends on N – exactly the problem we are trying to avoid. Instead, we propose a second MC estimator. Let $\tilde{x}_1, \ldots, \tilde{x}_K \overset{iid}{\sim} N(\mu, 1)$ be a single random sample of size K and define:

$$\widetilde{P_0(\mu)} = \frac{1}{K} \sum_{k=1}^{K} (1 - p(\tilde{x}_k))^T.$$

The posterior density can then be approximated by replacing $P_0(\mu)^{N-n}$ with:

$$\widetilde{Q(\mu)} = \left(\widetilde{P_0(\mu)}\right)^{N-n} = \left(\frac{1}{K}\sum_{k=1}^{K}(1-p(\tilde{x}_k))^T\right)^{N-n}.$$

This produces a biased but consistent estimator of the posterior density, but we conjecture that it maintains the overall properties of MCWM described by Andrieu and Roberts (2009). We believe that samples produced by the MCWM algorithm using $\widetilde{Q(\mu)}$ as an estimator of $P_0(\mu)^{N-n}$ will still approximate draws from the true posterior distribution for large enough K, though this remains to be proved.

A further advantage of the second MC estimator is that it allows the Gibbs update of N to be performed without further computation. Recall that the update of N depends only on $P_0(\mu)$ – exactly the value estimated in our MCWM update of μ. If μ' is accepted then we set $P_0(\widetilde{\mu^{(t+1)}}) = \widetilde{P_0(\mu')}$. Otherwise we set $P_0(\widetilde{\mu^{(t+1)}}) = \widetilde{P_0(\mu^{(t)})}$. Our full algorithm proceeds by:

A) Updating $\mu^{(t)}$ given $N^{(t)}$ via MCWM:

1) Propose $\mu' \sim q(\mu|\mu^{(t)})$

2) Compute MC estimates $\widetilde{P_0(\mu^{(t)})}$ and $\widetilde{P_0(\mu')}$, and the corresponding estimates $\widehat{\pi}_K(\mu^{(t)}|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$ and $\widehat{\pi}_K(\mu'|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})$.

3) Accept μ' and set $\mu^{(t+1)} = \mu'$ with probability $\hat{\alpha} = \min(1, \hat{r}(\mu^{(t)}, \mu'))$ where:

$$\hat{r}(\mu^{(t)}, \mu') = \frac{\widehat{\pi}_K(\mu'|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})}{\widehat{\pi}_K(\mu^{(t)}|N^{(t)}, \boldsymbol{X}^{obs}, \boldsymbol{Y}^{obs})} \cdot \frac{q(\mu^{(t)}|\mu')}{q(\mu'|\mu^{(t)})}.$$

Otherwise, set $\mu^{(t+1)} = \mu^{(t)}$.

B) Updating $N^{(t)}$ given $\mu^{(t+1)}$ via Gibb's sampling:

$$N^{(t+1)} \sim \text{Neg. Bin.}(n, 1 - P_0(\widetilde{\mu^{(t+1)}}))$$

2.3 Extensions

We propose two extensions of MCWM that seem to provide more efficient sampling for mark-recapture models. The first is to use related samples in computing the MC estimates of the posterior density in both the numerator and denominator of the Hastings ratio. Consider the simple model. The basic property of location-scale families can be used to generate $\tilde{x} \sim N(\mu, \sigma^2)$: if $\tilde{z} \sim N(0,1)$ then $\tilde{x} = \sigma\tilde{z} + \mu \sim N(\mu, \sigma^2)$. In our implementation of the MCWM algorithm, we use a single sample of K independent standard normal random variates to estimate both $P_0(\mu^{(t)})$ and $P_0(\mu^{'})$. Specifically, we generate $\tilde{z}_1, \ldots, \tilde{z}_K \overset{iid}{\sim} N(0,1)$ and define:

$$\widetilde{P_0(\mu^{(t)})} = \frac{1}{K}\sum_{k=1}^{K}(1 - p(\tilde{z}_k + \mu^{(t)}))^T \text{ and } \widetilde{P_0(\mu')} = \frac{1}{K}\sum_{k=1}^{K}(1 - p(\tilde{z}_k + \mu'))^T.$$

The advantage is that the MC samples used in the numerator and denominator of $\hat{r}(\mu^{(t)}, \mu')$ have the same quantiles with respect to their corresponding distributions. This ensures that extreme values do not occur in one of the MC samples alone and seems to improve mixing. The same procedure can also be applied using uniform random variates and the probability integral transformation if the distribution of x_i is not in a location-scale family.

The second modification we have tested is to use antithetic sampling in constructing the MC estimates. Instead of generating K distinct values from the normal distribution, we generate a random normal sample of size $K/2$ (assuming K is even), $\tilde{z}_1, \ldots, \tilde{z}_{K/2} \overset{iid}{\sim} N(0, 1)$, and then set $\tilde{z}_{K/2+k} = -\tilde{z}_k$, $k = 1, \ldots, K/2$. This induces negative correlation within the MC sample and reduces the variance of the MC estimator if the integrand is a monotone function of x (see for example Givens and Hoeting, 2012, pg. 187–188). This is true for the simple model above and for the model in Section 4 which treats β_0 and β_1 as unknown. Similar methods can also be applied for non-normal covariates and in higher dimensions. We refer to the MCWM algorithm combined with antithetic sampling as MCWM/AS.

3 Simulation Study

To demonstrate the properties of MCWM, we describe results from a small simulation study based on the simple model presented in Section 2.2. We assumed a population of $N = 1000$ individuals and $T = 5$ capture occasions. We generated 100 data sets each for two different values of μ. In the first scenario, we set $\mu = -1$ such that $E(p_i) = .30$ and $P_0(\mu) = .25$. In the second scenario, we set $\mu = -3$ such that $E(p_i) = .07$ and $P_0(\mu) = .73$.

Samples from the posterior distribution conditional on each simulated data set were generated via RJMCMC, DA, MCWM, and MCWM/AS. We also compared the effects of varying the size of the super-population for DA and the size of the MC sample for MCWM. We first ran RJMCMC for each data set and then applied DA with M equal to r times the largest value of N sampled during the RJMCMC algorithm for $r = 1, 2, 4$. Finally, we applied both MCWM and MCWM/AS with MC sample sizes of $K = 100, 500,$ and 1000.

Each algorithm depends on choices regarding the updaters of μ, N, and the augmented data (if applicable). We tried to implement the algorithms as would a relatively experienced user of MCMC. We applied Gibbs sampling steps when possible and otherwise used MH steps with standard proposal densities optimized through an adapting phase. Complete details of the different algorithms are provided in Table 1. All chains were started from the true parameter values to avoid effects of the initial values and were run for a total of 55,000 iterations with the first 5000 removed as burn-in. All code was written in `R` and vector calculations were used when possible. An `R` package containing code is available from the first author upon request.

For each of the two scenarios, we compared the efficiency of the different samplers and the accuracy of the estimated posterior summary statistics. Accuracy of the samplers was assessed by comparing the location and spread of the sampled values of N. Specifically, we compared the bias and mean-squared-error (MSE) of the posterior mean of N:

$$\text{Bias}(\hat{N}) = \sum_{s=1}^{100}(\hat{N}_s - 1000) \text{ and } \text{MSE}(\hat{N}) = \sum_{s=1}^{100}(\hat{N}_s - 1000)^2/100$$

where $\hat{N}_s$ represents the posterior mean estimated from the s^{th} simulation and the estimated posterior standard deviation of N. Efficiency of the samplers was assessed by comparing the effective number of samples for N generated per second (the effective sample size of N divided by the runtime of the chain). Simply comparing the runtime for the different algorithms is inappropriate because the samples are not independent. A chain that runs quickly but has high autocorrelation may be less efficient than a slower chain that mixes better. The effective sample size of an MCMC sample is the number of independent draws which would be needed to provide the same information about the posterior distribution. This value is estimated by fitting an autoregressive (AR) time series model to the sampled chain and then computing the integrated autocorrelation as described by Liu (2008, pg. 125–126) and implemented in the `coda` package in `R` (Plummer et al., 2006). Results are presented in Figures 1 and 2. Complete numerical results are also provided in the Supplementary Materials (Table S1).

Posterior summary statistics produced via RJMCMC and all variants of DA were almost identical for all of the 100 data sets in Scenario 1 ($\mu = -1$). The bias of the posterior means for RJMCMC and DA ranged between -0.3 to 0.2, and the MSE ranged from 63.4 to 65.2. MCWM and MCWM/AS also produced good estimates of the posterior means. The bias of these implementations was slightly higher with smaller values of K, but with $K = 1000$ the bias was less than 0.4 and the MSE was 63.4. However, MCWM tended to overestimate the posterior variance. Mean posterior standard deviations from RJMCMC and DA ranged between 22.3 and 22.6, and MCWM overestimated the posterior standard deviation by approximately 1.7 times when $K = 100$ and 1.1 times when $K = 1000$. However, the problem was almost completely resolved by the use of antithetic sampling. MCWM/AS overestimated the posterior standard deviation by approximately 1.1 times when $K = 100$ and almost not at all when $K = 1000$.

The clear advantage of both MCWM and MCWM/AS was the gain in efficiency. The runtimes for the different variants of MCWM and MCWM/AS were similar to the runtimes for RJMCMC and DA with $r = 1$, but the chains mixed much more quickly. Even with $K = 1000$, MCWM and MCWM/AS were approximately 3.5 times as efficient as the most efficient DA algorithm and more than 100 times as efficient as the RJMCMC algorithm. Antithetic sampling had little effect on these results. On average, MCWM/AS did run slightly faster than MCWM, but the small difference was offset by the change in effective sample size.

Results for Scenario 2 ($\mu = -3$) were qualitatively similar. The posterior summary statistics produced by RJMCMC and all variants of DA were close. Posterior means from these methods were biased by approximately 0.5% due to the influence of the selected prior for N which favors smaller values. Once again, MCWM overestimated the posterior mean of N when $K = 100$, and both MCWM and MCWM/AS also overestimated the posterior standard deviation for all values of K. However, the error was less than 2% on average for MCWM/AS with $K = 1000$. With $K = 500$, MCWM/AS continued to produce good estimates of the posterior mean and overestimated the standard deviation by only 4% on average.

Mean runtimes for DA and RJMCMC in Scenario 2 were between 1.2 and 1.7 times the mean runtimes in Scenario 1. In comparison, the mean runtimes of MCWM and MCWM/AS decreased slightly because the speeds of DA and RJMCMC depend on the upper bound on N, which increased from Scenario 1 to Scenario 2, while the speeds of MCWM and MCWM/AS depend on n, which decreased. Effective sample sizes for all algorithms decreased in Scenario 2, but MCWM and MCWM/AS were still more efficient than RJMCMC and all variants of the DA algorithm. With $K = 1000$, MCWM/AS was 22.0 times as efficient as RJMCMC and 13.0 times as efficient as the best version of DA. As before, reducing K to 500 affected the accuracy of the posterior summary statistics slightly but increased the efficiency even further so that MCWM/AS was 29.6 times as efficient as RJMCMC and 17.5 times as efficient as DA.

In summary, MCWM/AS with large values of K (500 or 1000) performed well in both scenarios. Posterior summary statistics were almost equal to those produced by DA and

RJMCMC, but MCWM/AS was much more efficient. Decreasing K reduced the accuracy of the estimated posterior summary statistics, in particular the posterior standard deviation, but led to a further increase in efficiency. It was surprising that RJMCMC had such low efficiency, and we discuss this result further in Section 5.

[Table 1 about here.]

[Figure 1 about here.]

[Figure 2 about here.]

4 Application

As an example of these methods, we analyzed data taken from a study of meadow voles (*Microtus pennsylvanicus*) conducted at the Patuxent Wildlife Research Center in 1981 and 1982 (Nichols et al., 1992). The experiment followed a robust design with 6 primary periods each comprising 5 capture occasions. We focus on the final primary period and assume that the population was closed over this time. The data from this period contain records of 77 voles of which 23 (30%) were captured once and 54 (70%) twice or more. The average number of captures per marked vole was 2.7. We consider the average observed body mass for each vole as a static individual covariate and ignore issues with censoring and rounding discussed by Schofield and Barker (2011).

The model we fit to this data is the same as the model described in Section 2.2, except that we treat all parameters as unknown. This includes abundance, N, the coefficients of the logistic model for p_i, β_0 and β_1, and the parameters of the normal distribution for x_i, μ

and σ^2. Once again, we specify a conjugate normal prior for μ and the improper Jeffrey's prior for N. For the remaining parameters, we selected the half t prior with three degrees of freedom for σ and independent t priors with three degrees of freedom for both β_0 and β_1. These represent weakly informative priors with most mass near 0 but also with heavy tails.

In this model, the probability that an individual is never captured is a function of μ, σ^2, β_0, and β_1. This requires that MC integration be used to estimate the posterior density in the update steps for each of these parameters. In our implementation, we update $\boldsymbol{\beta} = (\beta_0, \beta_1)'$ as a single unit, and so our algorithm requires three separate MCWM steps per iteration of the MCMC algorithm along with the Gibbs update of N.

As in the simulation study, we compared i) samples generated via MCWM and MCWM/AS with varying values of K, ii) samples from DA with varying values of r, and iii) samples from RJMCMC. We again implemented all algorithms using standard updating procedures: Gibb's sampling where possible and MH updates with standard proposals otherwise. The algorithms were again implemented in `R` and chains were run for a total of 500,000 iterations with a burn-in period of 50,000 iterations. All code is available from the first author. Plots of the results are provided in the top half of Figures 3 and 4. Numeric summaries are provided in the Supplementary Materials (Table S2).

Posterior summary statistics from all implementations were almost exactly identical. Even with $K = 25$, MCWM and MCWM/AS provided very accurate results. runtimes for the different implementations were also similar, except that MCWM and MCWM/AS both took significantly longer when K was large ($K = 1000$). Once again, the RJMCMC implementation mixed slowly and had much lower efficiency than the other algorithms. However, MCWM and MCWM/AS provided no advantage over DA. The best DA implementation ($r = 2$) was in fact 1.1 times more efficient that the best MCWM implementation (MCWM/AS with $K = 100$).

The MCWM approach is intended to address computational problems that arise with DA and RJMCMC when the proportion of individuals captured is small (n much less than N), and so we have repeated the analysis with a modified version of the meadow vole data constructed by artificially decreasing the capture probability for each marked individual. Specifically, we generated new data by 1) replicating the capture histories for each of the 77 marked voles 5 times, 2) subsampling the captures in the resulting histories with probability 0.2, and 3) removing histories with no remaining captures. The resulting data contained 159 histories with 122 (77%) individuals being captured once and only 39 (23%) twice or more. The average number of captures per marked individual was 1.3. Plots of the results are provided in the bottom half of Figures 3 and 4. Numeric summaries are provided in the Supplementary Materials (Table S2).

In this case, posterior means obtained from MCWM were comparable with the other methods but the posterior standard deviation was overestimated when K was small. This was corrected completely by MCWM/AS, and estimated posterior summary statistics obtained from MCWM/AS were indistinguishable from the other methods.

Once again, the advantage of MCWM is clear. Whereas the runtime of RJMCMC increased 1.4 times and the runtime of DA increased between 1.8 and 3.0 times depending on r, the runtime of both MCWM and MCWM/AS increased by less than 1.1 time for all values of K. As a result, MCWM and MCWM/AS with $K = 100$ were both approximately 2.5 times as efficient as RJMCMC and the fastest implementation of DA. Note that the efficiency of all of the algorithms, including MCWM and MCWM/AS, decreased significantly with the modified data. This simply reflects the fact that the autocorrelation of the Markov chains is higher when n is small.

[Figure 3 about here.]

[Figure 4 about here.]

5 Discussion

The examples presented in Sections 3 and 4 provide an initial assessment of MCWM for fitting mark-recapture models with heterogeneity. As expected, MCWM performed nearly as well as the other algorithms when most individuals were marked and was more efficient when the proportion of marked individuals was small. Not only was the runtime for MCWM smaller in these situations because the computational complexity depends on observed sample size, rather than the size of the population or super-population, but the chains produced by MCWM also mixed more quickly. The disadvantage is that MCWM samples from an approximation to the posterior distribution and the accuracy of the posterior summary statistics depends on the MCMC sample size (K). Posterior summary statistics will be biased if K is too small, but the algorithm will take a long time to run and sampling will be inefficient if K is too large. Selecting an appropriate value for K remains as an important question.

Although the examples presented involved scalar covariates, we intend these methods for modeling more complex data with high-dimensional covariates. When capture probabilities depend on a scalar covariate the probability that an individual from the population is never captured, P_0, could be computed with numerical quadrature (see Appendix 1 in the Supplementary Materials). Choquet and Gimenez (2010) and Gimenez and Choquet (2010) have used this approach to evaluate the likelihood for mark-recapture models with scalar individual random effects. However, quadrature methods with regular grids can be inefficient for computing integrals in high-dimensions essentially because the integrand may be close to zero at many of the grid points. In these cases, MC integration can be more efficient if the sampling distribution concentrates on the regions of the sample space where the integrand is non-zero Liu (2008, pg. 32). In future, we will apply MCWM to fit both closed and open population models with high-dimensional integrals, focusing primarily on data with time-varying, individual covariates as in Bonner and Schwarz (2006).

We believe that the methods presented will be most useful for modeling data from large populations in which the overall capture probability is low. Fitting these models with DA will require large super-populations and might lead to long runtimes. In these cases, MCWM may provide accurate inference in much shorter times allowing users to explore a range of models more easily. We also believe that MCWM could provide an alternative to DA and CDL methods used to model other complex ecological data (e.g., spatially explicit mark-recapture models (Royle et al., 2008) or distance sampling models including individual covariates (Royle et al., 2004)).

We will also investigate further modifications that might improve the accuracy or efficiency of MCWM. Using antithetic sampling within the MCWM steps improved the accuracy of posterior summary statistics significantly, and further gains may be made by incorporating more advanced MC methods. For example, importance sampling could be used to estimate the probability that an individual is never captured. Defining an appropriate important sampling distribution *a priori* will be difficult, but this could be chosen through an adaptive scheme. We also plan to explore two related algorithms that make use of MC integration within MCMC: the Grouped Independence Metropolis-Hastings (GIMH) algorithm (Beaumont, 2003; Andrieu and Roberts, 2009) and the Monte Carlo Metropolis-Hastings (MCMH) algorithm (Liang et al., 2010). Incredibly, both algorithms produce Markov chains that converge to the exact posterior distribution when the MC estimator of the posterior density (GIMH) or MH acceptance ratio (MCMH) is unbiasedAndrieu and Roberts (2009). Unfortunately the only unbiased estimator of the posterior density we have found requires K samples of size $N - n$ which reintroduces the dependence on N (see Section 2.2). Further study is needed to understand the properties of these algorithms if a biased but consistent estimator is used instead.

Finally, the simulation study raised new questions about the RJMCMC and DA algorithms. In particular, the efficiency of both algorithms improved in some cases when the amount of data augmentation increased. Consider the DA algorithm. Conventional wisdom has suggested that M be as small as possible (though it must be big enough not to restrict the posterior distribution of N). To avoid penalizing the DA algorithm by selecting an arbitrary value we originally set M equal to the largest value of N generated by the RJMCMC algorithm ($r = 1$). We later found that the efficiency of DA could be increased with a larger value of M as in simulation Scenario 1 with $r = 2$. Similarly, we were surprised to find that the efficiency of the RJMCMC algorithm was higher in Scenario 2, when only 25% of the population was captured, than in Scenario 1, when 75% of the population was captured.

The increase in the efficiency of DA seems to occur because the chains mix better and the effective sample size is larger when M is bigger. We believe that this occurs because a larger super-population can generate more populations of size N. This allows the composition of the population to change more freely when unobserved individuals are drawn from the super-population on each MCMC iteration, and in turn allows for larger changes in the model parameters. Although the runtime increases when M increases, the computational cost of the vector calculations used to update $\boldsymbol{z}$ and $\boldsymbol{X}_{\boldsymbol{M}}^{\boldsymbol{miss}}$ in our implementation of DA increases slowly with M and may be offset if the mixing improves sufficiently. Further research is needed to determine if the same result occurs with other software and if there is an optimal value for the size of the super-population.

The problem is harder to address for RJMCMC because the amount of augmentation is not pre-determined. More efficient variants of RJMCMC might be implemented using different proposal distributions for N. Our proposal distribution is based on King and Brooks (2008), except that we adapted the width of the uniform distribution to produce an acceptance rate near 50%. In some cases, the proposal distribution was very limited and N could not change by more than 1 or 2 on each iteration. Skewed distributions which allow for occasional large jumps might improve the efficiency, and further research is needed to identify optimal proposal distributions.

Acknowledgments

We thank Dr. Jim Nichols for allowing us to use the Patuxent meadow vole data in our example. SJB and MRS were partially funded by the NSF KY-EPSCOR fund (NSF Grant

No. 0814194). MRS was also partially funded by NSF Grant No. 0934516.

References

Alho, J. M. (1990). Logistic regression in capture-recapture models. *Biometrics*, 46(3):623–35.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–60.

Bonner, S. J. and Schwarz, C. J. (2006). An extension of the Cormack-Jolly-Seber model for continuous covariates with application to *Microtus pennsylvanicus*. *Biometrics*, 62(1):142–9.

Choquet, R. and Gimenez, O. (2010). Towards built-in capturerecapture mixed models in program E-SURGE. *Journal of Ornithology*, 152(S2):625–639.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the {EM} algorithm. *Journal of Royal Statistical Society, Series B (Methodological)*, 39(1):1–38.

Dupuis, J. A. (1995). Bayesian estimation of movement and survival probabilities from capture- recapture data. *Biometrika*, 82(4):761–772.

Gelman, A. E., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). {*B*}*ayesian Data Analysis*. Chapman and Hall, 2nd edition.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). Introducing Markov chain Monte Carlo. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov chain Monte Carlo in Practice*, chapter 1, pages 1–19. Chapman and Hall, New York.

Gimenez, O. and Choquet, R. (2010). Individual heterogeneity in studies on marked animals using numerical integration: capturerecapture mixed models. *Ecology*, 91(4):951–957.

Gimenez, O., Crainiceanu, C. M., Barbraud, C., Jenouvrier, S., and Morgan, B. J. T. (2006). Semiparametric regression in capture-recapture modeling. *Biometrics*, 62(3):691–8.

Givens, G. H. and Hoeting, J. A. (2012). *Computational statistics*. Wiley.

Huggins, R. M. (1989). On the statistical analysis of capture experiments. *Biometrika*, 76(1):133–140.

King, R. and Brooks, S. P. (2008). On the Bayesian estimation of a closed population size in the presence of heterogeneity and model uncertainty. *Biometrics*, 64(3):816–24.

King, R., Brooks, S. P., Morgan, B. J. T., and Coulson, T. (2006). Factors influencing soay sheep survival: a Bayesian analysis. *Biometrics*, 62(1):211–220.

Liang, F., Liu, C., and Carroll, R. J. (2010). *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. John Wiley and Sons Ltd., West Sussex, UK.

Link, W. A. (2013). A cautionary note on the discrete uniform prior for the binomial N. *Ecology*. In press.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Verlag.

McDonald, T. L. and Amstrup, S. C. (2001). Estimation of population size using open capture-recapture models. *Journal of Agricultural, Biological, and Environmental Statistics*, 6(2):206–220.

Nichols, J. D., Sauer, J. R., Pollock, K. H., and Hestbeck, J. B. (1992). Estimating transition probabilities for stage-based population projection matrices using capture-recapture data. *Ecology*, 73:306–312.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). Coda: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11.

Pollock, K. H. (2002). The use of auxiliary variables in capture-recapture modelling: an overview. *J. Appl. Statist*, 29:85–102.

Royle, J. A. (2009). Analysis of capture-recapture models with individual covariates using data augmentation. *Biometrics*, 65(1):267–274.

Royle, J. A., Dawson, D. K., and Bates, S. (2004). Modeling abundance effects in distance sampling. *Ecology*, 85(6):1591–1597.

Royle, J. A. and Dorazio, R. M. (2012). Parameter-expanded data augmentation for Bayesian analysis of capturerecapture models. *Journal of Ornithology*, 152(S2):521–537.

Royle, J. A., Dorazio, R. M., and Link, W. A. (2007). Analysis of multinomial models with unknown index using data augmentation. *Journal of Computational and Graphical Statistics*, 16(1):67–85.

Royle, J. A., Young, K. V., and Young, V. (2008). A hierarchical model for spatial capture-recapture data. *Ecology*, 89(8):2281–2289.

Schofield, M. R. and Barker, R. J. (2011). Full open population capture-recapture models with individual covariates. *Journal of Agricultural, Biological, and Environmental Statistics*, 16(2):253–268.

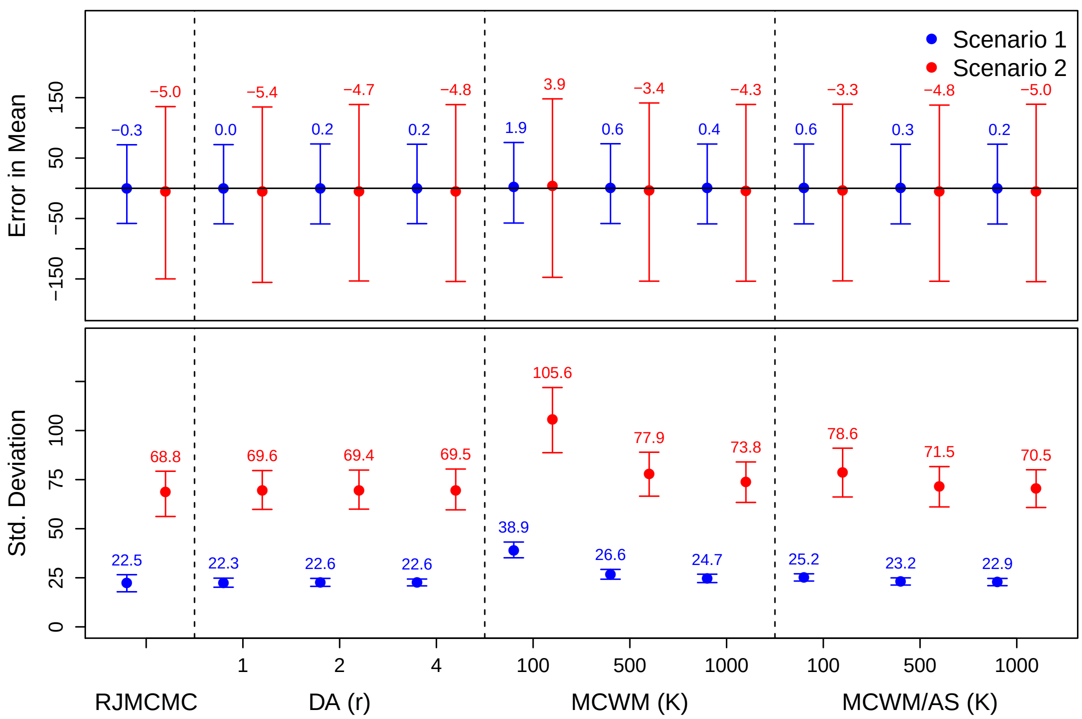

Figure 1: Simulation Results 1 – Posterior Summaries. Distributions of the error in the posterior mean (top) and the posterior standard deviations (bottom) of N for Scenario 1 (blue symbols) and Scenario 2 (red symbols) for the different MCMC implementations. Points in each plot represent the mean value over all 100 simulated data sets. These values are also provided numerically. Error bars connect the largest and smallest values over the 100 simulated data sets.

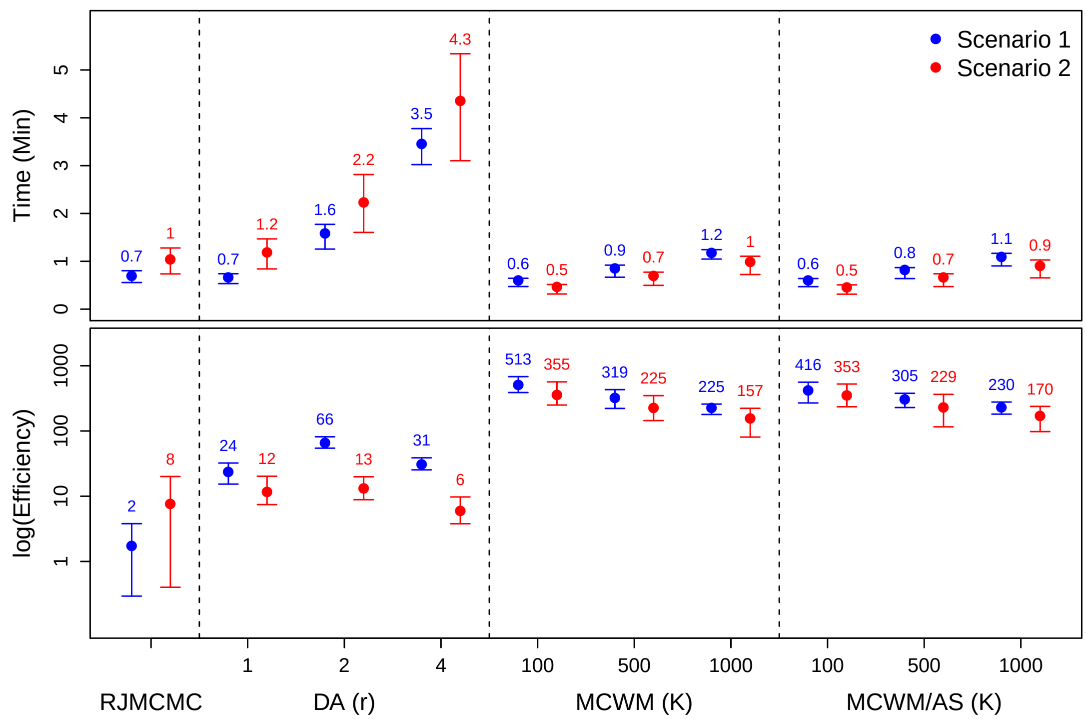

Figure 2: Simulation Results 2 – Efficiency. Comparisons of the runtime in minutes (top) and log efficiency for sampling N (effective sample size/second) of the different MCMC implementations for Scenario 1 (blue symbols) and Scenario 2 (red symbols). The points represent the mean runtime/efficiency over the 100 replicate data sets. These values are also provided numerically. The error bars extend to the limits of the runtime/efficiency observed over the 100 simulated data sets.

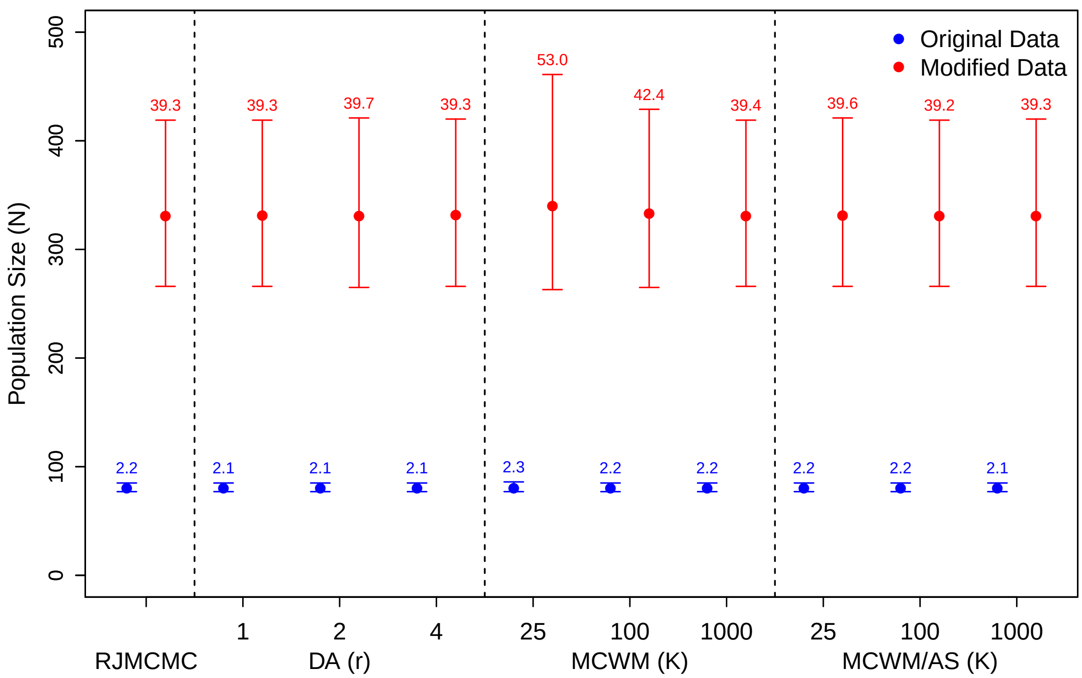

Figure 3: Application Results 1 – Posterior Summaries. Comparison of the posterior distribution for the original meadow vole data (blue symbols) and the modified data (red symbols) for the different MCMC implementations. The estimated posterior mean for each implementation is represented by the point with 95% credible interval represented by the error bar. Values above the error bars indicate the estimated posterior standard deviation.

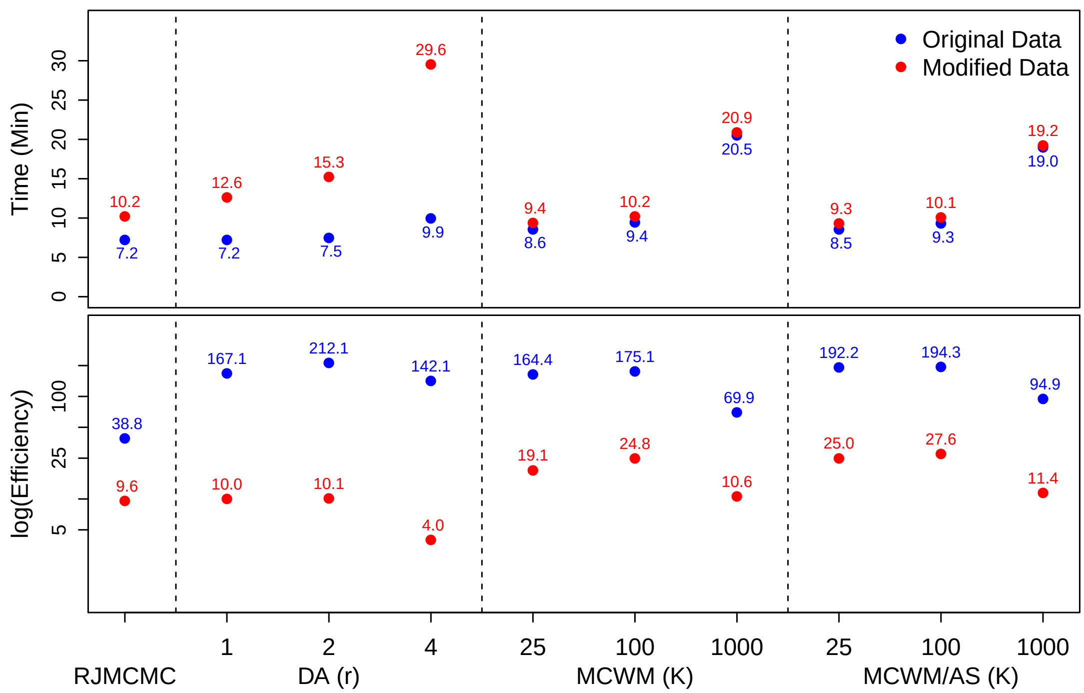

Figure 4: Application Results 2 – Efficiency. Comparison of the runtime (top) and efficiency (bottom) of the different MCMC implementations in the analysis of the original data (blue symbols) and modified data (red symbols). The top plot compares the time taken in minutes. The bottom plot compares the efficiency for sampling N (effective sample size/second).

RJMCMC

Parameter	Update Method
μ	Gibbs step: $\mu^{(t+1)} \sim N\left(\frac{\nu^2}{\sigma^2}\sum_{i=1}^{N} x_i, \nu^2\right)$, where $\nu^2 = \left(\frac{1}{\tau_\mu^2} + \frac{N}{\sigma_x^2}\right)^{-1}$
N	RJ step with proposal: $$N' \sim U\{N^{(t)} - r, \ldots, N^{(t)} - 1, N^{(t)} + 1, \ldots, N^{(t)} + r\}$$
x_i	MH step with proposal $x_i \sim N(\mu, 1)$

DA

Parameter	Update Method
μ	Gibbs step: $\mu^{(t+1)} \sim N\left(\left(\frac{\sigma_x^2}{\tau_\mu^2} + M\right)^{-1}\sum_{i=1}^{M} x_i, \left(\frac{1}{\tau_\mu^2} + \frac{M}{\sigma_x^2}\right)^{-1}\right)$
ψ	Gibbs step: $\psi \sim \text{Beta}(\alpha + \sum_{i=1}^{M} z_i, \beta + M - \sum_{i=1}^{M} z_i)$
x_i	MH step with proposal $x_i \sim N(\mu, 1)$
z_i	Gibbs step: $z_i \sim \text{Bernoulli}\left(\frac{\psi(1-p_i)^T}{(1-\psi)+\psi(1-p_i)^T}\right)$

MCWM

Parameter	Update Method
μ	MCWM step with proposal $\mu' \sim N(\mu^{(t)}, \xi_\mu^2)$.
N	Approximate Gibbs step: $N \sim \text{Neg. Bin}(n, 1 - \widehat{P_0(\mu)})$

Table 1: Implementations choices for the variants of the MCMC algorithms. The three sections of the table describe the updates for each parameter in RJMCMC (top), DA (middle), and MCWM (bottom). The implementation of MCWM/AS was the same as MCWM except that antithetic sampling was used to estimate the posterior density in the MCWM update of μ.