Using Hierarchical Centering to Facilitate a Reversible Jump MCMC Algorithm for Random Effects Models

Using hierarchical centering to facilitate a reversible jump MCMC algorithm for random effects models

C. S. Oedekoven[a,*], R. King[b], S. T. Buckland[a], M. L. Mackenzie[a],
K. O. Evans[c] and L. W. Burger, Jr.[c]

[a] *Centre for Research into Ecological and Environmental Modelling,*
School of Mathematics and Statistics, University of St Andrews,
The Observatory, Buchanan Gardens, St Andrews, KY16 9LZ, UK Tel.: +44 1334 461826 Fax: +44 1334 461 800
[b] *School of Mathematics, University of Edinburgh,*
James Clerk Maxwell Building, The King's Buildings, Peter Guthrie Tait Road, Edinburgh, UK, EH9 3FD
[c] *Department of Wildlife, Fisheries & Aquaculture,*
Mississippi State University,
Box 9690, Mississippi State, MS 39762, USA
**email: cso2@st-andrews.ac.uk*

Abstract

Hierarchical centering has been described as a reparameterisation method applicable to random effects models. It has been shown to improve mixing of models in the context of Markov chain Monte Carlo (MCMC) methods. A hierarchical centering approach is proposed for reversible jump MCMC (RJMCMC) chains which builds upon the hierarchical centering methods for MCMC chains and uses them to reparameterize models in an RJMCMC algorithm. Although these methods may be applicable to models with other error distributions, the case is described for a log-linear Poisson model where the expected value λ includes fixed effect covariates and a random effect for which normality is assumed with a zero-mean and unknown standard deviation. For the proposed RJMCMC algorithm including hierarchical centering, the models are reparameterized by modelling the mean of the random effect coefficients as a function of the intercept of the λ model and one or more of the available fixed effect covariates depending on the model. The method is appropriate when fixed-effect covariates are constant within random effect groups. This has an effect on the dynamics of the RJMCMC algorithm and improves model mixing. The methods are applied to a case study of point transects of indigo buntings where, without hierarchical centering, the RJM-

[1] The data and R code for the case study are provided in the annexes of the electronic version of this manuscript.
[2] * cso2@st-andrews.ac.uk
[3] * http://creem2.st-andrews.ac.uk/
[4] * http://coedekoven.wix.com/cornelia-oedekoven

CMC algorithm had poor mixing and the estimated posterior distribution depended on the starting model. With hierarchical centering on the other hand, the chain moved freely over model and parameter space. These results are confirmed with a simulation study. Hence, the proposed methods should be considered as a regular strategy for implementing models with random effects in RJMCMC algorithms; they facilitate convergence of these algorithms and help avoid false inference on model parameters.

Keywords: combined likelihood, "Metropolis Hastings", point transect sampling, random effects, reparameterisation.

---

1. Introduction

For Bayesian analyses, for a given model, the posterior distribution of the parameters is formed by combining the likelihood of the data with the prior distributions of the parameters. A Markov chain Monte Carlo (MCMC) algorithm is often used to sample from this posterior distribution to obtain inference on the parameters of interest. In the presence of model uncertainty, the posterior distribution can be extended to be defined jointly over both parameter and model space. This posterior distribution is often explored using the reversible jump Markov chain Monte Carlo (RJMCMC) algorithm (Green, 1995). However, the art of setting up an RJMCMC algorithm can be challenging on multiple levels. The objective is generally to construct a chain that moves freely between models, efficiently exploring model and parameter space simultaneously.

The RJMCMC algorithm entails iteratively updating the parameters conditional on the model (i.e. within-model move) and then updating the model (and corresponding model parameters) conditional on the current parameters (i.e. between-model move). Mixing problems for the within-model moves are often due to high autocorrelation within the constructed Markov chain. Improvements for mixing within a given model have been investigated in the framework of MCMC with the aim of reducing posterior correlations and increasing the effective sample size by reparameterisation. In this context, Browne (2004) and Browne et al. (2009) have shown that hierarchical centering (first described by Gelfand, Sahu, and Carlin, 1995) can significantly reduce the autocorrelation within the MCMC algorithm. The use of hierarchical centering in the presence of random effects refers to exchanging the zero-mean of the random effect component, typically assumed to be of normal form, with a model consisting of an intercept and one or more fixed effect covariates.

This will be described in detail in section 2. Papaspiliopoulos, Roberts, and Sköld (2007) investigated the circumstances when hierarchical centering performs well in comparison to noncentering for MCMC algorithms.

Other methods for improving mixing of an MCMC algorithm include parameter expansion, which refers to augmenting the model with additional parameters to form an expanded model (Browne, 2004). The original model is embedded in the expanded one and parameters from the original model can be constructed with parameters from the expanded model. Vines, Gilks, and Wild (1995) describe a method of reparameterisation for random effects models called *sweeping* which is suitable also for models with multiple sets of random effects in a generalized linear mixed model (*glmm*) framework. The idea consists of adding the mean of the random effect coefficients to the intercept of the fixed effects while subtracting the same quantity from each random effect coefficient.

For the between-model move in an RJMCMC algorithm (the RJ step), the current model is updated by proposing to move to an alternative model (with given parameters) and accepting this move with some probability. Mixing problems for these between-model moves may arise for multiple reasons, e.g. due to difficulties in finding proposal distributions and updating procedures that produce suitable acceptance probabilities. Besides careful pilot-tuning of proposal distributions, several methods for improving the acceptance rate for the reversible jump step have been proposed. For example, Green and Mira (2001) proposed delayed rejection, where after initial rejection a second attempt to jump is made with samples generated from a new distribution that may depend on the rejected proposal. Brooks, Giudici, and Roberts (2003) assumed a family for the proposal distribution, where the proposal parameters are chosen to maximize (in some form) the acceptance probability. Al-Awadhi, Hurn, and Jennison (2004) demonstrated that increasing acceptance probabilities may be achieved by using a secondary Markov chain with a fixed number of steps that serves to move the value of an RJMCMC proposal closer to a mode before calculating the acceptance probability for the proposed move. Papathomas, Dellaportas, and Vasdekis (2011) proposed that model mixing for generalized linear models may be improved by using proposal densities that draw samples from parameter subspaces of competing models. Forster, Gill, and Overstall (2012) used the Laplace approximation to integrate out the random effects and orthogonal projections of the current linear predictor onto the proposed linear predictor to produce effective proposals for

*glmm*s.

While these previous approaches describe strategies to improve the acceptance rate for RJ steps in general, they can be quite complex to implement. We propose an approach using hierarchical centering that is relatively straightforward to implement for random/mixed effect models. A particular problem that one may encounter with random effect models is that the random effect coefficients may begin absorbing the effect of one or more fixed effect covariates if these are not present in the model at times during the Markov chain. The inclusion of such effects into the model may then be unlikely as they are already accounted for within the random effects. We will demonstrate below that using hierarchical centering provides a simple way of reparameterising the model that will prevent this problem and improve the between-model mixing.

Hierarchical centering was initially described by Gelfand, Sahu, and Carlin (1995) as a method to improve convergence for mixed models using MCMC methods. Here we extend the ideas to improve mixing in an RJMCMC algorithm. Although our methods may be applicable to models with other error distributions, we consider the case for a log-linear Poisson model with fixed effects and a normally distributed random effect, where the overall likelihood combines the Poisson likelihood for each observation and the normal density for each random effect coefficient. We demonstrate how the Poisson likelihoods and the normal densities are affected differently during a proposal to add a covariate for a regular RJMCMC algorithm and one including hierarchical centering.

We demonstrate the improved model mixing using a case study of point transects of indigo buntings (*Passerina cyanea* L.). Point transects are a form of distance sampling (Buckland et al., 2001) where, in addition to the number of detections during the counts, distance from the point to each detection is collected. This allows estimation of the average detection probabilities at the point and adjustment of counts for imperfect detection. To study the effect of establishing conservation buffers along margins of agricultural fields on density of several species of conservation interest, pairs of points were set up at the edge of fields in a number of states in the USA. These pairs of points consisted of one point on a treatment field and one on a nearby control field without a buffer and these pairs will be referred to as sites in the following. Counts were repeated 1–4 times in each year 2006–2007. We use a combined likelihood including the likelihoods for the detection function and the log-linear Poisson model where counts are adjusted for imperfect detection within the search area around the point (Oedekoven et al., 2014). A random effect for site is included in

the Poisson model to accommodate correlated counts between different sites.

In the following we describe how to implement hierarchical centering for RJMCMC, describe the effects on the dynamics of the algorithm, and present updating methods for the RJ step using hierarchical centering (section 2). We then apply the methods to our case study (section 3) and confirm our results with a simulation study (section 4) and discuss our findings (section 5).

2. Hierarchical Centering

The hierarchical centering described in this paper refers to mixed effect models where a normal distribution is assumed for the random effect. Other distributions may be assumed for the random effect (e.g. Komárek and Lesaffre, 2008) to which these methods can be applied but we focus on the normal distribution for simplicity. We describe the case for a *glmm* with a Poisson error structure, suitable e.g. for fitting a model to correlated count data from repeated measurements. In the following we denote the different groups for the random effect with subscript j and the repeated measurements within the individual groups with subscript r. Here, the expected value λ_{jr} is modelled via a log-link function with a common intercept, β_0 and random effect coefficients b_j for groups *j* are included for which normality is assumed. For a mixed effect model without hierarchical centering, the random effect is incorporated into the model under the assumption of a global zero-mean and unknown standard deviation, σ_b, i.e. $b_j \sim N\left(0, \sigma_b^2\right)$ (e.g. Bates, 2009). Let us assume we have a set of K covariates for $k = 1, ..., K$, x_k (and associated coefficients, β_k) that can be incorporated as fixed effects. The expected value for the full model including all covariates may then be expressed as:

$$\lambda_{jr} = \exp\left(\beta_0 + \sum_{k=1}^{K} x_{kjr}\beta_k + b_j\right), \;\; b_j \sim N\left(\mu_j = 0, \sigma_b^2\right), \tag{1}$$

where the x_{kjr} are the measured covariate values corresponding to the rth observation of the response of group j. While all potential models include the intercept and the random effect, different models included in the RJMCMC algorithm correspond to the combinations of covariates present in the model (i.e. non-zero β_k values). During a between-model move (the RJ step) of an RJMCMC algorithm using this scenario, the proposal to delete or add one (or more) of the covariates alters the formula for λ_{jr} while the distribution for the random effects terms b_j remains the same

(see Appendix A for details on the RJ step).

Let us now assume that one covariate, say x_1, was measured at the group level, i.e. values for all repeated measurements for this covariate within a given group were the same, which allows us to use x_1 for hierarchical centering. In hierarchical centering, the mean of the random effect is modelled using a combination of the intercept β_0 and one or more covariates that are "pulled from" the λ_{jr} model from (1) (Gelfand et al., 1995). In the case that the intercept and covariate x_1 are used for centering, the full model from (1) becomes:

$$\lambda_{jr} = \exp\left(\sum_{k=2}^{K} x_{kjr}\beta_k + b_j\right),\ \ b_j \sim N\left(\mu_j = \beta_0 + x_{1j}\beta_1, \sigma_b^2\right). \tag{2}$$

Note that we omitted the subscript r for covariate x_1 in (2) since we assume that the measured values for this covariate were the same for all observations in group j. The proposal to delete or add x_1 from the model during the RJ step of the RJMCMC algorithm involves altering the distribution for b_j, while the proposal to delete or add any other covariates remains the same as before in (1) (altering the formula for λ_{jr}).

In the case that all k covariates were measured at the group level, all covariates may be included in the centering and the full model from (1) becomes:

$$\lambda_{jr} = \exp\left(b_j\right),\ \ b_j \sim N\left(\mu_j = \beta_0 + \sum_{k=1}^{K} x_{kj}\beta_k, \sigma_b^2\right). \tag{3}$$

Again, we omitted the subscript r for the covariates in the model for μ_j in (3). In (3), it could be omitted from λ_{jr} as well, as there are no covariates in the λ_{jr} model (or the μ_j model) that may vary between different observations within the same group. However, we keep it for simplicity in the following equations. In this scenario, the formula for λ_{jr} remains unchanged during the proposals to delete or add any of the covariates, while the distribution for b_j changes for each proposed model move.

We note that it is essential that only those covariates are included in the centering (i.e. x_1 in (2) or x_k with *k=1,...,K* in (3)) that have the same measured value for all observations within a group (Browne et al., 2009). We refer to a group in terms of the grouping unit for the random effect where grouping should occur to account for intra-group dependence (Davison, 2003). All

observations belonging to the same group j are modelled with the same random effect coefficient b_j in the equations above.

The proposed hierarchical centering is only applicable if for at least one covariate, measured values for the respective covariate are the same within a group. If, for example, the grouping unit for a study is *site*, then the covariate *state* (the geographical governed entity) can be included in the centering as each site only belongs to one state and all repeated observations for a site belong to the same state. Conversely, *Julian day* could not be included as values will likely vary between repeated measurements. As long as this condition holds, any combination of covariates may be included.

Hierarchical centering relies on the fact that the random effect coefficients pick up the effect of the covariates included in the centering (given that they have an effect) as they are updated during the within-model move of each iteration of the RJMCMC. Running separate MCMC algorithms (without between-model moves) on the full models from (1), (2) or (3) should result in nearly identical summary statistics for the covariates if the chain was run long enough (since all Markov chains have the same stationary distribution), although mixing might be different for these different parameterisations. However, when including the between-model moves in an RJMCMC algorithm, mixing problems can become more severe, potentially leading to different summary statistics - due to lack of convergence - and hence potentially to the wrong conclusions. Here, convergence and, hence, obtaining correct results may depend on which scenario and initial starting values were used. If, e.g. under the scenario of (1), the random effect coefficients absorb the effect of covariate x_1, the chain may get "stuck" in models that do not include x_1. For the scenarios of (2) and (3), moves to models including covariate x_1 would be favoured if the random effect coefficients absorbed the effect of x_1 as then the coefficients will be closer to their modelled means. We will show below that this is due to the fact that here different parts of the likelihood are affected by a proposed model move compared to (1).

2.1. Effects of hierarchical centering on RJMCMC dynamics

Using either one of the models for λ_{jr} from above ((1), (2), or (3)), the likelihood of the log-linear Poisson model, $L_n(\boldsymbol{\beta}, \sigma_b)$, with a normally distributed random effect may be formulated as

(modified from McCulloch and Searle, 2001):

$$L_n(\boldsymbol{\beta}, \sigma_b) = \prod_{j=1}^{J} \left(\prod_{r=1}^{R_j} \frac{(\lambda_{jr})^{n_{jr}} \exp(-\lambda_{jr})}{n_{jr}!} \times \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(b_j - \mu_j)^2}{2\sigma_j^2} \right) \right), \tag{4}$$

where vector $\boldsymbol{\beta}$ contains the coefficients for covariates included in the models and n_{jr} are the observed measurements of the response. The indices $j = 1, 2, 3, ..., J$ represent the groups for the random effect and $r = 1, 2, 3, ..., R_j$ indices for the different measurements taken for the *j*th group. Hence for each group of observations, j, the probability of observing n_{jr} under the log-linear Poisson model with expected value of λ_{jr} is multiplied for all observations within that group, which is then multiplied by the normal density of the random effect coefficient b_j. The only coefficients that influence both parts of this likelihood, i.e. the Poisson likelihood for the observations and the normal densities, are the random effect coefficients, regardless of which scenario is used from the previous section.

Consider now, that we use this likelihood as part of calculating the acceptance probabilities for updating the model as well as the fixed and random effect coefficients in an RJMCMC algorithm (e.g. Oedekoven et al., 2014). Both the Poisson likelihood and the normal densities are higher if the observed value of the response or the random effect coefficients are closer to their respective means (λ_{jr} or μ_j, respectively). Hence, combining what we know from (1) - (4), it is evident that the Poisson likelihoods will improve if the variation that is not accounted for by the fixed effect coefficients is picked up by the random effect coefficients (which – as well as the fixed effect coefficients of the current model – are updated during the within-model move). On the other hand, the normal densities will return higher values for random effect coefficients close to their mean values.

Intuitively, one may think that a problem arises for a between-model move (using models from (1)) when a covariate, say x_1, may have an effect but is not included in the current model. Then, the random effect coefficients may begin to absorb this effect and in this manner, adjust the value for λ_{jr} to improve the likelihood. This may result in a "tug-of-war" between the Poisson likelihood trying to adjust the coefficients in such a manner that the effect of x_1 is accounted for and, on the other hand, the normal densities trying to keep the coefficients close to their mean, i.e. zero for (1)). This will typically also result in an inflated random effect standard deviation since the

random effect coefficients are replacing some unexplained variability attributable to x_1. If this has indeed occurred, an acceptance of x_1 into the model during a between-model move proposal may become very unlikely as its effect is already accounted for by the random effect coefficients. Hence, during a proposal to add x_1, the new model with x_1 will create inferior λ_{jr}. These will then return decreased likelihood values even if the randomly drawn value(s) for x_1 would produce a larger likelihood under circumstances before the effect has been absorbed by the random effect coefficients. In other words, the values of the random effect coefficients are dependent on the model. A strategy to account for this could be to jointly update the coefficient value(s) for covariate x_1 and the values for the random effect coefficients. However, this complicates the RJ step involving more complex proposal distributions.

Alternatively, this issue may be addressed using hierarchical centering since proposing to add x_1 using either (2) or (3) into the model will not change λ_{jr} (and the Poisson likelihood). Here, the random effect coefficients absorb the effects of the covariates included in the model (given they have an effect) within the mean of the random effect distribution (in addition to the intercept β_0). Using (2) this would be only covariate x_1; using (3) this would be covariates x_k with $k = 1, 2, 3..., K$. The only part of the likelihood that is affected when updating this/these covariate(s) for within-model and between-model moves are the normal densities from (4). It is likely that, on average, the normal densities improve for the individual random effect coefficients as these will on average be closer to their assumed mean. As λ_{jr} remains the same, likelihood values returned by the Poisson part of (4) remain the same (which also increases the speed of calculating the acceptance probability for the RJ step since only the normal densities need to be evaluated).

2.2. RJ updating methods using hierarchical centering

To demonstrate how to implement hierarchical centering, we use a simple example where during the between-model move of iteration $t + 1$ we propose to include covariate x_1 into an intercept-only model, say model *m*. Suppose that at iteration t the current state of the chain is model m, where $\lambda_{jr} = \exp(b_j)$ with $b_j \sim N(\mu_j = \beta_0, \sigma_b^2)$ from (3) (although if x_1 is the only covariate available, $K = 1$ and (2) and (3) are equivalent). During iteration $t + 1$ we propose to move to model m' by adding covariate x_1. Hence, model m' is defined as $\lambda'_{jr} = \exp\left(b'_j\right)$ with $b'_j \sim N\left(\mu'_j = \beta'_0 + x_{1j}\beta'_1, \sigma_b'^2\right)$. For simplicity, let us assume that covariate x_1 represents a cat-

egorical covariate with only two levels where the first level is absorbed in the intercept β'_0 and the second level has an associated coefficient β'_1; hence, x_1 is either 0 for the first level or 1 for the second level. We note that these methods also apply in the case that the covariate used for centering has more than two factor levels. Let us further assume that all measurements within a group j belong to the same level of x_1 and that, for simplicity, we have 200 groups where groups $j = 1, ..., 100$ belong to the first level of x_1 and groups $j = 101, .., 200$ belong to the second level of x_1. We use the identity function as the bijective function (King et al., 2010):

$$u'_0 = \beta_0, \quad \beta'_0 = u_0, \quad \beta'_1 = u_1 \tag{5}$$

and draw samples u from the respective proposal distributions for the parameters β'_0 and β'_1. See Appendix A for further details.

In the following, we describe two different ways for implementing the RJ step. The difference between them lies in the definition of the proposal distributions for the new parameters for the between-model move, and, hence, should only have an influence on the acceptance probability of this move. The second approach (Section 2.2.2) uses more information compared to the first (Section 2.2.1) and should, on average, return higher acceptance rates for this move. Either method should not have an influence on estimated posterior summary statistics of the parameters in the final model given that the chain had an adequate burn-in.

2.2.1. Hierarchical centering using predefined proposal distributions

For this method, we define proposal distributions for the coefficients β_0, β'_0 and β'_1. If, for example, normal proposal distributions are used, we define the proposal distributions for coefficients β'_1 as $\beta'_1 \sim N\left(\mu'_1, {\sigma'_1}^2\right)$, for some predefined μ'_1 and σ'_1. Equivalently, the normal proposal distributions for the intercepts β_0 and β'_0 are defined as $\beta_0 \sim N\left(\mu_0, {\sigma_0}^2\right)$ and $\beta'_0 \sim N\left(\mu'_0, {\sigma'_0}^2\right)$ (for some predefined μ_0, σ_0, μ'_0 and σ'_0).

2.2.2. Hierarchical centering using updated proposal distributions

Here, the mean μ_0 of the proposal distribution for the global intercept β_0 of model m and the means μ'_0 and μ'_1 of the proposal distributions for the coefficients β'_0 and β'_1 of model m' are updated before the RJ step during each iteration of the RJMCMC algorithm. To update μ_0 at iteration

$t+1$, we take the overall mean $\bar{b}_j^t$ of the current values of all random effect coefficients b_j^t (i.e. $\beta_0^{t+1} \sim N(\mu_0^{t+1} = \bar{b}_j^t, \sigma_0^2)$ including groups $j = 1, ..., 200$). To update the μ_1' at iteration $t+1$, we take the mean $\bar{b}_j'^t$ of the random effect coefficients from iteration *t* belonging to the second level of covariate x_1. Hence, we have $\beta_1'^{t+1} \sim N(\mu_1'^{t+1} = \bar{b}_j'^t, \sigma_1'^2)$ only including groups $j = 101, ..., 200$. To update μ_0' at iteration $t+1$, we take the mean $\bar{b}_j'^t$ of all random effect coefficients belonging to the first level of covariate x_1 (i.e. groups $j = 1, ..., 100$).

3. Case study: point transects of indigo buntings

3.1. The data

To establish the success of planting herbaceous buffers around agricultural fields in several South-eastern and Midwestern US states, point transect surveys were conducted from a large number of randomly selected fields during the breeding season (May−July) of 2006−2007 in each participating state (Fig. Appendix B, Oedekoven et al., 2013). Survey points on control fields of the same agricultural use and located within 1−3km were surveyed concurrently. Each pair of adjacent points from a treated and control field was considered a site. Points were located at the edge of the fields. Observers recorded all male indigo buntings (all singles) detected either visually or aurally during a 10-minute count at each point in one of five predetermined distance intervals (0−25, 25−50, 50−100, 100−250, 250−500 and >500m). It is assumed that indigo buntings distribute themselves evenly within and in the various possible habitats adjacent to the field. Only those sites that were surveyed at least once in each survey year were included in the analysis. These 446 sites were located in nine states: Georgia, Iowa, Illinois, Kentucky, Missouri, Mississippi, Ohio, South Carolina and Tennessee.

3.2. Methods

As the models from (1) to (3) assume perfect detection on the plot, we needed to supplement these with a model to adjust counts for imperfect detection. We used the methods described in Oedekoven et al. (2014): a detection function was fitted to the frequency of detections in each distance bin. This detection function was used to estimate the effective area, ν (the area beyond which as many birds were seen as were missed within, Fig. Appendix B), which was incorporated into the log-linear Poisson model for the counts as an offset (Buckland et al., 2001). The full model

consisted of the likelihood component for the detection function and the likelihood component for the counts (see Oedekoven et al., 2014 for details). In addition, we extended the count model to include a subscript p to denote the two points at each site. With the offset included, the full model without hierarchical centering from (1) became:

$$\lambda_{jpr} = \exp\left(\beta_0 + \boldsymbol{x'_{1j}}\boldsymbol{\beta_1} + \sum_{k=2}^{K} x_{kjpr}\beta_k + b_j + \ln(\nu)\right), \; b_j \sim N\left(\mu_j = 0, \sigma_b^2\right). \tag{6}$$

Site was used as the grouping factor for the random effect. Available covariates were *state* ($\boldsymbol{x_1}$, a factor with nine levels), *year* (x_2, factor with two levels: 2006 and 2007, corresponding to $x_2 = 0$ and $x_2 = 1$, respectively), *Julian day* (x_3, discrete with observed integers ranging from 142 to 211) and *type* (x_4, factor with two levels: control ($x_4 = 1$) or treatment ($x_4 = 1$) plot). Factor covariate *state* is represented by a vector $\boldsymbol{x_{1j}}$ of length 8 either with eight entries zero for observations from state Georgia – as the coefficient of the baseline state is absorbed in the intercept – or with seven entries zero, and one 1, indicating which state site j was in, and $\boldsymbol{\beta_1}$ is a column vector of eight coefficients. Note that similar to (2) we omitted the subscripts r and p for covariate $\boldsymbol{x_1}$ since the values for this covariate were the same for all observations in group j. Furthermore, we did not include a subscript for the effective area ν as, for simplicity, we only considered global detection functions, i.e. without stratification or covariates in the detection model. Hence, given a model and parameter value(s) for the detection function, estimates of the effective area ν were the same for all counts. As *state* was the only covariate with consistent values for all measurements within a given site, we were limited to using only one covariate within the hierarchical centering (i.e. corresponding to (2)). With hierarchical centering using the *state* covariate, $\boldsymbol{x_1}$, the full model from (2) became:

$$\lambda_{jpr} = \exp\left(\sum_{k=2}^{K} x_{kjpr}\beta_k + b_j + \ln(\nu)\right), \; b_j \sim N\left(\mu_j = \beta_0 + \boldsymbol{x'_{1j}}\boldsymbol{\beta_1}, \sigma_b^2\right). \tag{7}$$

To estimate parameters of both the detection function ($\boldsymbol{\theta}$) and the count model ($\boldsymbol{\beta}, \sigma_b$) in one step, we combined the likelihood components pertaining to the respective models using the combined likelihood, $L_{n,y}(\boldsymbol{\beta}, \sigma_b, \boldsymbol{\theta}) = L_{yG}(\boldsymbol{\theta}) L_n(\boldsymbol{\beta}, \sigma_b | \boldsymbol{\theta})$ described by Oedekoven et al. (2014). In comparison to (4), $L_n(\boldsymbol{\beta}, \sigma_b | \boldsymbol{\theta})$ is conditional on detection function parameters $\boldsymbol{\theta}$ when including the

effective area as an offset in (6) or (7). The data contained $J = 446$ sites. R_j, the maximum number of visits to a site, ranged from 2 to 8 between sites as each site was visited 1-4 times in each of the two survey years. As each site contained two points, we extended (4) accordingly:

$$L_n(\boldsymbol{\beta}, \sigma_b|\boldsymbol{\theta}) = \prod_{j=1}^{446} \left(\prod_{p=1}^{2} \prod_{r=1}^{R_j} \frac{(\lambda_{jpr})^{n_{jpr}} \exp(-\lambda_{jpr})}{n_{jpr}!} \times \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(b_j - \mu_j)^2}{2\sigma_j^2} \right) \right). \quad (8)$$

As distances were recorded in intervals (rather than exact distances), the likelihood for the detection function component, $L_{yG}(\boldsymbol{\theta})$ was defined as the multinomial likelihood where f_i represents the probability that a detected animal is in the *i*th distance interval (for details on calculating the f_is see Appendix B):

$$L_{yG}(\underline{\theta}) = \left(\frac{n!}{\prod_{i=1}^{I} n_i!} \right) \prod_{i=1}^{I} f_i^{n_i}. \quad (9)$$

Here, *n* represents the total number of detected animals and n_i the number of animals detected in the *i*th distance interval. As detection probabilities generally dropped below 0.1 beyond 100m, we limited the analysis to the three innermost distance intervals (0–25, 25–50, 50–100m).

For the detection models, we considered the half-normal and hazard-rate key functions as the two (non-nested) model options (Buckland et al., 2001). For the count model, we considered all possible combinations of the covariates *year*, *type*, *Julian day* and *state*. We ran two different analyses on the same data. For the first analysis we used "regular" RJMCMC methods with a global zero-mean random effect (as shown in (6)) which we refer to as the global zero-mean analysis (GZM).

For the second analysis we implemented hierarchical centering by pulling the intercept β_0 and covariate *state* from the λ_{jpr} model and included them in the model for the random effect mean (as shown in (7)). This analysis will be referred to as HC in the following. We used predefined proposal distributions for all parameters. These were the same for both analyses (see Table B.1).

Prior model probabilities were equal and the identity function similar to (5) used for the bijective function of any proposed move. For both analyses, we placed the same set of uniform priors on the parameters (Table B.1).

For each analysis, the chain was started from the most parsimonious models: the half-normal

detection function and a count model containing the fixed effect intercept and a random effect for site. We ran 200 000 iterations for each analysis, the first 20 000 were considered as the burn-in phase. The effective sample size was calculated for each parameter in the preferred model using the function *effectiveSize* from the R package *coda*. We express it as the effective sample size per 1000 iterations that the chain was in the preferred model to make this quantity comparable between the results of different methods.

3.3. Results

The preferred detection model was the hazard-rate function with posterior probability of 1.00 for both analyses (Table B.2). Estimated probabilities for the count models differed between the methods. For GZM, the preferred count model included the covariates *type* and *Julian day* with probability 0.85. The alternative model included the additional covariate *year* and was selected during the remaining 15% of the iterations. The covariate *state* was never included in any of the models for this method. By contrast, all models included *state* for HC. The preferred model included *type*, *Julian day* and *state* (0.95 probability) and the second most preferred model included *type*, *Julian day*, *year* and *state* (0.05 probability).

While the probabilities of being in the model were similar for the covariates *year*, *Julian day* and *type* between the two analyses, the probability of *state* being in the model was 0.00 for GZM and 1.00 for HC. To investigate further, we used a range of different initial starting values and models to assess convergence. In particular, when we initialised the chain so that *state* was in the initial model for the GZM analysis, the posterior probability for *state* was 1.00. Repeated simulations provided the same output with *state* not being updated in GZM. Hence, for GZM the resulting model probabilities were conditional on the model that the chain was started with. In contrast, consistent results were obtained for the HC analysis, irrespective of initial values or initial model choice of the Markov chain.

Summary statistics for the parameters resulting from both GZM analyses (started with and without *state*) and the HC analysis are given in Table B.3. Means and 95% credible intervals (CRI) were nearly identical between all methods for the parameters of the hazard-rate detection function. Means and 95% CRIs were also similar for the count model parameters between the three methods, given that the parameters were in the model. Although means for parameter *Julian day*

varied, CRIs overlapped between all three methods. The exception was the random effect standard deviation of the count model which was very different for GZM started without *state* compared to the other two analyses. The mean was larger for GZM without *state* and CRIs did not overlap those of the other two methods. This was likely due to the random effects coefficients absorbing the *state* effect.

We refrain from including the GZM without *state* analysis in the comparison of effective sample sizes as here, due to non-convergence, the posterior distribution differed from the other two analyses (GZM with *state* and HC). The effective sample sizes for detection function parameters were similar between all the GZM with *state* and HC analyses (Table B.4). Effective sample sizes for count model parameters were generally smaller for HC compared to GZM with *state* except for the random effect standard deviation and the intercept. It was notable that the effective sample sizes for the *state* coefficients were consistently at least two times but up to over 12 times larger for GZM with *state* compared to HC. The only notable increase in effective sample size from GZM with *state* to HC was for the random effects standard deviation with 2.46 for GZM with *state* and 8.54 for HC.

4. Simulation study

The following simulation study was used to investigate whether our proposed methods would consistently improve model mixing. In particular, for a covariate with nested random effects that was part of creating the pattern in the response variable, we investigated whether posterior model probabilities would differ between hierarchical centering and regular RJMCMC methods. Using (2), we simulated 300 data sets of approximately 500 observations each that were similar to our case study. The response variable followed a Poisson distribution for which the expected value λ_{jr} was modelled as a function of a linear term, say *Julian day* x_2, and random effects coefficients, b_j for the jth site. The b_j were simulated using a factor covariate with five levels, say *state* $\boldsymbol{x_1}$ (with four associated coefficients $\boldsymbol{\beta_1}$ randomly drawn from a uniform distribution, $U(-2.6, 0.8)$, during each simulation and the coefficient of the first level absorbed in the intercept β_0), to model the mean μ_j of their normal distribution, $b_j \sim N\left(\mu_j = \beta_0 + \boldsymbol{x'_{1j}}\boldsymbol{\beta_1}, \sigma_b^2\right)$ and the random effects standard deviation $\sigma_b = 0.7$. Sites were nested within states with $25 - 35$ sites per state and repeat observations ($2 - 6$, subscript r) per site. We also created a dummy variable, a factor covariate

with eight levels which was not part of the model for generating the response. Similar to $\boldsymbol{x_1}$, this dummy variable had constant levels within each random effect group. However, the levels of the dummy variable to which random effects groups were attributed were chosen at random and did not match the pattern for attributing random effects groups to levels of $\boldsymbol{x_1}$.

Each data set was analysed using two different approaches equivalent to GZM without *state* and the HC methods above. The former refers to "regular" RJMCMC methods with a global zero-mean random effect (as shown in (1)). The latter refers to hierarchical centering methods where the intercept β_0, *state* and the dummy variable were included in the model for the random effect mean (as shown in (3)).

The RJMCMC analyses for each data set were initiated with the models for λ_{jr} and μ_j that only contained the intercept and random effects coefficients between the two models combined and the chains for both analysis methods had the same initial coefficient values. Both approaches used the same proposal distributions for new parameter values, the same mechanism for updating the model, i.e. proposing to add or delete covariates depending on whether it was currently in the model (including the dummy variable), and the same MH algorithm for updating parameter values. Each analysis included 100 000 iterations where the first 10 000 were considered burn-in.

For the GZM without *state* analysis, posterior probabilities of *state* being in the model were 0 for all 300 data sets. By contrast, posterior probabilities of *state* being in the model for the HC analysis were on average 0.94 (95% CRI = $\{0.62,1.00\}$) across all 300 data sets. The random effects standard deviation was generally overestimated for those models without *state*, i.e. those iterations of the HC analysis where *state* was not in the model (posterior distribution mean 0.92, 95% CRI = $\{0.69,1.24\}$) and for all models from the GZM analysis (0.96, $\{0.71,1.32\}$). By contrast, for those models with *state* from the HC analysis, the posterior distribution of this parameter (0.66, $\{0.49,0.88\}$) was more accurate with a mean closer to the known true value 0.7. The marginal posterior probability that the dummy variable was included in the model was zero for all 300 data sets and both analysis methods.

5. Discussion

The purpose of incorporating random effects in count models is generally to model variation that is otherwise unaccounted for. When using RJMCMC methods, the danger exists that the

random effect coefficients account for too much of the variation and prevent the inclusion of a fixed effect covariate into the model – a problem that is not limited to the linear predictor for the Poisson distribution. We demonstrated this case with our GZM analysis that was initiated without *state* in the model. Due to poor mixing (between models) leading to lack of convergence, the covariate *state* was never selected. This would have led to incorrect inference as the sampled values are not from the posterior distribution due to poor mixing. In addition, for this analysis the resulting random effect standard deviation was much larger compared to the HC analysis of the same data. Both these findings, the poor model mixing and inflated random effects standard deviation for the GZM analysis, were confirmed by our simulation study.

For the HC analysis of the case study, the model was also initiated without *state* but revealed posterior probabilities of *state* being in the model of 1.00. Furthermore, the mean and 95% CRI of the random effects standard deviation were smaller compared to the GZM without *state* analysis. Both these findings were again confirmed by our simulation study. For both analyses of the case study that were initiated without *state* in the model, GZM and HC, the random effect coefficients absorbed the effect of the *state* covariate. For GZM, this prevented the inclusion of this parameter into the model. For HC, this favoured the inclusion of *state* into the model as here this covariate was part of the model for the random effect mean. Here, the chain was able to explore models with *state* as a covariate due to improved mixing between models.

Unsurprisingly, implementing hierarchical centering had little effect on the remaining covariates in the model as these were not involved in the centering. However, we could not confirm the findings of Browne (2004), that implementing hierarchical centering would improve the effective sample size for the covariate involved in the centering. He compared the effective sample sizes for the same covariate in two different MCMC chains, one with hierarchical centering and one without. For our case study, effective sample sizes for coefficients involved in the centering were mostly larger for GZM with *state* compared to HC except for the intercept and the random effect standard deviation where, using hierarchical centering, the effective sample size increased 3.47-fold.

Overall we showed that implementing hierarchical centering in the context of RJMCMC algorithms improves mixing between models and, hence, improves the inference on model parameters. For our case study, summary statistics for covariates not involved in the centering were nearly

identical between the GZM and the HC analyses. However, inference on the *state* covariate using the GZM analysis could potentially have led us to believe falsely that this covariate had no effect on densities of indigo buntings.

Acknowledgements

The National CP-33 Monitoring Program was funded by the Multistate Conservation Grant Program (Grant MS M-1-T), which is supported by the Wildlife and Sport Fish Restoration Program and managed by the Association of Fish and Wildlife Agencies and US Fish and Wildlife Service. Further support was provided by the US Department of Agriculture (USDA) Farm Service Agency and USDA Natural Resources Conservation Service Conservation Effects Assessment Project. Collaborators included the AR Game and Fish Commission, GA Department of Natural Resources (DNR), IL DNR/Ballard Nature Center, IN DNR, IA DNR, KY Department of Fish and Wildlife Resources/KY Chapter of The Wildlife Society, MS Department of Wildlife, Fisheries and Parks, MO Department of Conservation, NE Game and Parks Commission, NC Wildlife Resources Commission, OH DNR, SC DNR, TN Wildlife Resources Agency, TX Parks and Wildlife Department, Southeast Quail Study Group and Southeast Partners In Flight. The first author was supported by a studentship jointly funded by the University of St Andrews and EPSRC, through the National Centre for Statistical Ecology (EPSRC grant EP/C522702/1), with subsequent funding from EPSRC/NERC grant EP/1000917/1. We thank Prof. William Browne, University of Bristol, for inspiring the presented methods and for commenting on a draft manuscript.

Appendix A. RJMCMC algorithm

In general, the posterior distribution $\pi\left(\delta_m, m|x\right)$ is given as the distribution encompassing the joint posterior distribution of models and parameters (Green, 1995; King et al., 2010) with:

$$\pi\left(\delta_m, m|x\right) \propto L(x|\delta_m, m)p(\delta_m|m)p(m). \tag{A.1}$$

Here, $L(x|\delta_m, m)$ is the probability density function of the data x conditional on model *m* with current parameter values δ_m, $p(\delta_m|m)$ is the prior probability for model parameters δ_m conditional on the chain being in model *m*, and $p(m)$ is the prior probability of model *m*.

Suppose that we propose to move from model m with parameters δ_m to model m' with parameters $\delta'_{m'}$ during the between-model move (RJ step) of an RJMCMC algorithm. We define u and u' as random samples from some proposal distribution for the respective parameters. To transform parameters δ_m into $\delta'_{m'}$ we use a bijective function which may have the form $(\delta'_m, u') = g(\delta_m, u)$. Then, the acceptance probability is given by $\min(1, A)$ where A can be expressed as:

$$A = \frac{\pi(\delta'_{m'}, m'|x)P(m|m')q'(u')}{\pi(\delta_m, m|x)P(m'|m)q\,(u)} \left| \frac{\partial g(\delta_m, u)}{\partial(\delta_m, u)} \right|. \tag{A.2}$$

$P(m'|m)$ is the probability of proposing to move to model m' given that the chain is in model *m*, $q(u)$ and $q'(u')$ are the proposal densities of u and u'. $\left| \frac{\partial g(\delta_m, u)}{\partial(\delta_m, u)} \right|$ is the Jacobian.

For the within-model move (the MH step) of the RJMCMC algorithm we use a random walk single-update Metropolis-Hastings algorithm (Hastings, 1970; Metropolis et al., 1953).

Appendix B. The detection function component

To calculate the offset for (6) and (7), we used the probability density function of observed distances, $f(y) = \pi(y)g(y)/\int_0^w \pi(y)g(y)dy$, where w is the truncation distance (Buckland et al., 2001). The function describing the distribution of birds is given for points by $\pi(y) = 2y/w^2$ and the detection function is given by $g(y)$. We included two detection functions as model options in the RJMCMC algorithm, the half-normal ($g(y) = \exp\left(-y^2/2\sigma^2\right)$) and the hazard-rate ($g(y) = 1 - \exp\left(-(y/\sigma)^{-\tau}\right)$). When using interval distance data (as opposed to exact distance measurements), f_i is defined as the probability that a detected animal is in the *i*th interval which is delineated by the cutpoints c_{i-1} and c_i and is given by:

$$f_i = \frac{\int_{c_{i-1}}^{c_i} f\,(y)\,dy}{\int_0^w f\,(y)\,dy}, \tag{B.1}$$

where the truncation distance, *w* corresponds to the outermost cutpoint. The f_i feed into the likelihood component given in (9). $g(y)$ is also used to calculate the effective area, which for points is given by $\nu = 2\pi \int_0^w yg(y)dy$.

Literature cited

Al-Awadhi, F., Hurn, M., Jennison, C., 2004. Improving the acceptance rate of reversible jump MCMC proposals. Statistics & Probability Letters 69, 189–198.

Bates, D., 2009. Computational methods for mixed models. R package version 0.999375-31. Tech. rep., http//lme4.r-forge.r-project.org/.

Brooks, S. P., Giudici, P., Roberts, G. O., 2003. Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. Journal of the Royal Statistical Society B 65(1), 3–55.

Browne, W. J., 2004. An illustration of the use of reparameterisation methods for improving MCMC efficiency in crossed random effect models. Multilevel Modelling Newsletter 16, 13–25.

Browne, W. J., Steele, F., Golalizadeh, M., Green, M. J., 2009. The use of simple reparameterizations to improve the efficiency of Markov chain Monte Carlo estimation for multilevel models with applications to discrete time survival models. Journal of the Royal Statistical Society A 172 (3), 579598.

Buckland, S. T., Anderson, D. R., Burnham, K. P., Laake, J. L., Borchers, D. L., Thomas, L., 2001. Introduction to Distance Sampling. Oxford University Press.

Davison, A. C., 2003. Statistical Models. Cambridge University Press.

Forster, J. J., Gill, R. C., Overstall, A. M., 2012. Reversible jump methods for generalised linear models and generalised linear mixed models. Statistics and Computing 22 (1), 107–120.

Gelfand, A. E., Sahu, S. K., Carlin, B. P., 1995. Efficient parametrisations for normal linear mixed models. Biometrika 82 (3), 479–488.

Green, P. J., 1995. Reversible Jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82(4), 711–732.

Green, P. J., Mira, A., 2001. Delayed rejection in reversible jump Metropolis-Hastings. Biometrika 88(4), 1035–1053.

Hastings, W. K., 1970. Monte Carlo sampling methods using Markov Chains and their applications. Biometrika 57(1), 97–109.

King, R., Morgan, J., Gimenez, O., Brooks, S., 2010. Bayesian Analysis for Population Ecology. CRC Press.

Komárek, A., Lesaffre, E., 2008. Generalized linear mixed model with a penalized Gaussian mixture as a random effects distribution. Computational Statistics and Data Analysis 52, 3441–3458.

McCulloch, E. C., Searle, S. R., 2001. Generalized, Linear, and Mixed Models. John Wiley & Sons, Inc.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., 1953. Equations of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1091.

Oedekoven, C. S., Buckland, S. T., Mackenzie, M. L., Evans, K. O., Burger, L. W., 2013. Improving distance sampling: accounting for covariates and non-independency between sampled sites. Journal of Applied Ecology 50(3), 786–793.

Oedekoven, C. S., Buckland, S. T., Mackenzie, M. L., King, R., Evans, K. O., Burger, L. W., 2014. Bayesian methods for hierarchical distance sampling models. Journal of Agricultural, Biological, and Environmental Statistics 19 (2), 219–239.
URL http://dx.doi.org/10.1007/s13253-014-0167-0

Papaspiliopoulos, O., Roberts, G. O., Sköld, M., 2007. A general framework for the parametrization of hierarchical models. Statistical Science 22 (1), 5973.

Papathomas, M., Dellaportas, P., Vasdekis, V. G. S., 2011. A novel reversible jump algorithm for generalized linear models. Biometrika 98 (1), 231–236.

Vines, S. K., Gilks, W. R., Wild, P., 1995. Fitting multiple random effects models. Tech. rep., MRC Biostatistics Unit, Cambridge.

List of Figures

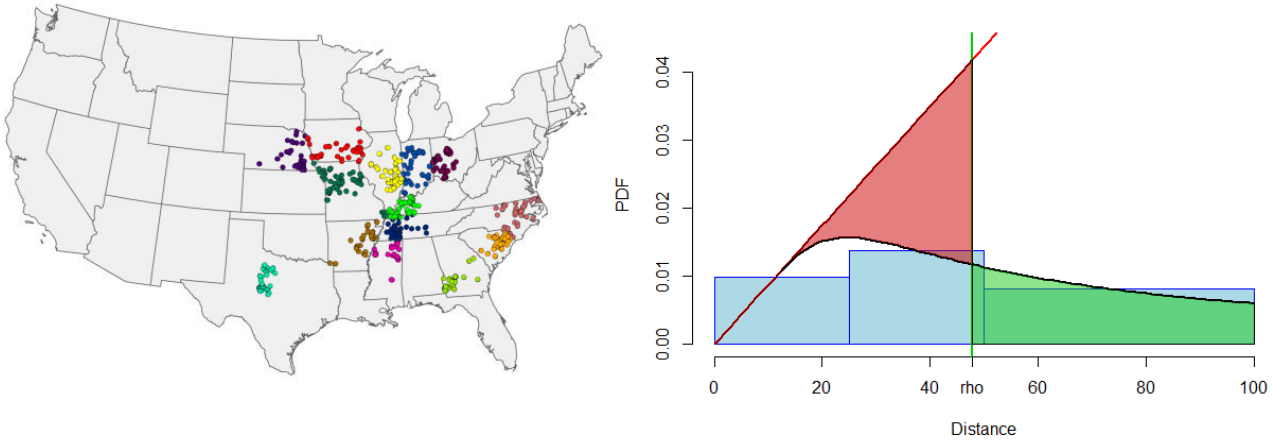

Figure B.1: Left: Distribution of surveys conducted as part of the CP-33 Monitoring Program between 2006-20011 (source: http://www.fwrc.msstate.edu/bobwhite/monitoring/index.asp). Right: frequency of detections in the three distance bins (0–25, 25–50, 50–100m) as rescaled blue histogram bars; probability density of observed distances (PDF) using means from the posterior distribution of parameters of the hazard-rate detection function (see Table B.3, black line); the slope of the red line is the slope of the PDF at distance zero; rho is the radius of the effective area ν; the red polygon represents the proportion of birds missed within rho and is equal in size to the green polygon which represents the proportion of birds detected between rho and the truncation distance w of 100m (Buckland et al., 2001). See Appendix B for more details.

List of Tables

Table B.1: Means and standard deviations (SD) of normal proposal distributions for model parameters as well as lower and upper boundaries for uniform prior distributions for model parameters. HN and HR refer to the half-normal and the hazard-rate detection functions respectively. We note that the random effect standard deviation and the intercept for the count model were always in the model.

Parameters	**Mean**	**SD**	**Lower**	**Upper**
Detection Function Parameters				
Scale HN:	37	2	10	99
Scale HR:	28	2	10	99
Shape HR:	2	1	1	10
Count Model Parameters				
Random effect standard deviation	–	–	0	1
Intercept:	–	–	-20	-7
Year level: 2007	0.05	0.2	-1	1
Type level: Treated	0.3	0.1	0	1
Julian Day:	0.0055	0.003	-0.1	0.1
State level: IL	0.4	0.5	-2.5	2.5
State level: IN	0.3	0.5	-2.5	2.5
State level: KY	0.7	0.5	-2.5	2.5
State level: MO	0	0.5	-2.5	2.5
State level: MS	0.5	0.5	-2.5	2.5
State level: OH	0	0.5	-2.5	2.5
State level: SC	0.2	0.5	-2.5	2.5
State level: TN	0.8	0.5	-2.5	2.5

Table B.2: Posterior model probabilities for the analyses of the indigo bunting data. Shown are results from the GZM analysis (global zero-mean for the random effect) and results from the HC (hierarchical centering) analysis. Both analyses were started without *state* in the initial model.

Analysis	**GZM**	**HC**
Detection Model		
CDS: Hazard-rate key	1.000	1.000
Count Model		
Type + JD	0.851	0.000
Year + Type + JD	0.149	0.000
Type + JD + State	–	0.946
Year + Type + JD + State	–	0.054

Table B.3: Mean and 95% credible intervals for models with highest posterior support from the analyses of the indigo bunting data. Results are from the analyses GZM (global zero-mean) started without *state*, GZM started with *state* and HC (hierarchical centering) started without *state*. For models with *state*, the *state* level GA is absorbed in the intercept.

Analysis	**GZM without *state***	**GZM with *state***	**HC**
Detection Model Parameters			
Scale HR σ	28.20	28.21	28.05
	(25.03,31.25)	(24.77,31.23)	(25.00,31.04)
Shape τ	2.08	2.08	2.08
	(1.92,2.26)	(1.91,2.26)	(1.92,2.25)
Count Model Parameters			
Random effect standard deviation σ_b	0.77	0.58	0.51
	(0.65,0.91)	(0.49,0.68)	(0.45,0.57)
Intercept β_0	-10.62	-10.76	-10.44
	(-11.21,-10.13)	(-11.25,-10.29)	(-10.97,-10.01)
Type level: Treated β_4	0.31	0.31	0.30
	(0.24,0.37)	(0.24,0.37))	(0.24,0.37)
Julian Day β_3	0.008	0.006	0.004
	(0.006,0.012)	(0.003,0.009)	(0.002,0.007)
State level: IL $\beta_{1_{IL}}$	-	0.88	0.97
	-	(0.54,1.23)	(0.63,1.32)
State level: IN $\beta_{1_{IN}}$	-	0.70	0.79
	-	(0.36,1.06)	(0.45,1.14)
State level: KY $\beta_{1_{KY}}$	-	1.16	1.24
	-	(0.84,1.50)	(0.90,1.57)
State level: MO $\beta_{1_{MO}}$	-	0.27	0.35
	-	(-0.02,0.58)	(0.04,0.67)
State level: MS $\beta_{1_{MS}}$	-	1.12	0.97
	-	(0.76,1.49)	(0.64,1.31)
State level: OH $\beta_{1_{OH}}$	-	0.40	0.39
	-	(0.08,0.71)	(0.07,0.72)
State level: SC $\beta_{1_{SC}}$	-	0.66	0.68
	-	(0.31,1.02)	(0.32,1.04)
State level: TN $\beta_{1_{TN}}$	-	1.31	1.38
	-	(0.98, 1.65)	(1.04,1.72)

Table B.4: Effective sample sizes per 1000 iterations that the chain was in the respective preferred model for model parameters from the analyses of the indigo bunting data: GZM (global zero-mean) with *state* in the initial model and HC (hierarchical centering).

Parameter	**GZM with *state***	**HC**
Detection Model		
Scale HR	5.04	4.99
Shape	6.18	5.92
Count Model		
Random effect standard deviation	2.46	8.54
Intercept	0.57	1.06
Type Treatment	71.15	64.07
Julian Day	0.58	0.26
State IL	4.15	0.64
State IN	3.63	0.69
State KY	2.58	0.42
State MO	3.38	0.28
State MS	3.30	0.57
State OH	3.54	0.39
State SC	4.41	0.69
State TN	3.74	0.52