

A comparison of strategies for Markov chain Monte Carlo computation in quantitative genetics

Rasmus Waagepetersen*

Department of Mathematical Sciences
Aalborg University, DK-9220 Aalborg, Denmark
`rw@math.aau.dk`

Noelia Ibánñez-Escriche

IRTA

Avda. Rovira Roure, E-25198 Lleida, Spain

Daniel Sorensen

Department of Genetics and Biotechnology
Danish Institute of Agricultural Sciences
P.O. Box 50, DK-8830 Tjele, Denmark

Short title: Strategies for Markov chain Monte Carlo computation.

Abstract

In quantitative genetics, Markov chain Monte Carlo (MCMC) methods are indispensable for statistical inference in non-standard models like generalized linear models with genetic random effects or models with genetically structured variance heterogeneity. A particular challenge for MCMC applications in quantitative genetics is to obtain efficient updates of the high-dimensional vectors of genetic random effects and the associated covariance parameters. We discuss various strategies to approach this problem including reparameterization, Langevin-Hastings updates, and updates based on normal approximations. The methods are compared in applications to Bayesian inference for three data sets using a model with genetically structured variance heterogeneity

Key words: Langevin-Hastings, Markov chain Monte Carlo, normal approximation, proposal distributions, reparameterization.

1 Introduction

Given observations of a trait and a pedigree for a group of animals, the basic model in quantitative genetics is a linear mixed model with genetic random effects. The correlation matrix of the genetic random effects is determined by the pedigree and is typically very high-dimensional but with a sparse inverse. Maximum likelihood inference and Bayesian inference for the linear mixed model are well-studied topics [16]. Regarding Bayesian inference, with appropriate choice of priors, the full conditional distributions are standard distributions and Gibbs sampling can be implemented relatively straightforwardly.

The assumptions of normality, linearity, and variance homogeneity are in many cases not valid. One may then consider generalized linear mixed models where the genetic random effects enter at the level of the linear predictor. San Cristobal-Gaudy *et al.* [15] proposed another extension of the linear mixed model introducing genetic random effects influencing the log residual variances of the observations thereby producing a genetically structured variance heterogeneity. Considerable computational problems arise when abandoning the standard linear mixed model. Classical maximum likelihood and Bayesian inference is complicated since it is not possible to evaluate explicitly the likelihood function. Using conventional Gibbs sampling to approximate the posterior distribution is difficult since the full conditional distributions are not anymore of standard forms.

The aim of this paper is to discuss strategies to obtain efficient Markov chain Monte Carlo (MCMC) algorithms for non-standard models of the kind mentioned in the previous paragraph. Such algorithms may be used either to approximate posterior distributions or to generate samples for importance sampling approximations of the likelihood [6]. In particular we focus on the problem of constructing efficient updating schemes for the high-dimensional vectors of genetic random effects. We review the methodological background and discuss the various algorithms in the context of the heterogeneous variance model. Apart from being a model of great interest in its own right, this model has proved to be a hard test for MCMC methods. We compare the performances of the different algorithms when applied to three real datasets which differ markedly both in size and regarding the inferences concerning the genetic covariance parameters.

Section 2 discusses general strategies for obtaining efficient MCMC algo-

rithms while Section 3 considers these strategies in the specific context of the San Cristobal-Gaudy *et al.* [15] model. Section 4 presents results of applying two MCMC schemes to data sets with pig litter sizes, rabbit litter sizes, and snail weights. Some concluding remarks are given in Section 5.

2 MCMC strategies for high-dimensional problems

We initially discuss MCMC strategies in a rather general framework where given the vector of random additive genetic effects $\mathbf{a} = (a_1, \dots, a_M)$ and a parameter vector $\boldsymbol{\beta}$, the vector \mathbf{y} of observed traits follows some density $f(\mathbf{y}|\mathbf{a}, \boldsymbol{\beta})$. As usual in quantitative genetics, \mathbf{a} is assumed to be zero mean normal with covariance matrix $\sigma_a^2 \mathbf{A}$, where \mathbf{A} is the additive genetic relationship matrix that reflects the family structure, typically known, and σ_a^2 is the additive genetic variance. In the following (apart from Section 2.5) we assume known σ_a^2 and $\boldsymbol{\beta}$ and focus on MCMC strategies for sampling from the posterior

$$p(\mathbf{a}|\mathbf{y}) \propto f(\mathbf{y}|\mathbf{a}, \boldsymbol{\beta})p(\mathbf{a}|\sigma_a^2) \quad (1)$$

where we for notational convenience omit $\boldsymbol{\beta}$ and σ_a^2 on the left hand side. An algorithm for sampling \mathbf{a} can typically easily be extended with updates of the lower dimensional quantities $\boldsymbol{\beta}$ and σ_a^2 in order to sample the full posterior distribution of $(\mathbf{a}, \sigma_a^2, \boldsymbol{\beta})$ (Section 2.5). An introduction to MCMC can be found e.g. in [16].

2.1 The Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm generates a Markov chain $\mathbf{a}^1, \mathbf{a}^2, \dots$ as follows. Given the current value \mathbf{a}^i of the Markov chain, a proposal \mathbf{a}^{prop} is generated from a proposal density $q(\mathbf{a}^{\text{prop}}|\mathbf{a}^i)$. With probability

$$\min\left\{1, \frac{p(\mathbf{a}^{\text{prop}}|\mathbf{y})q(\mathbf{a}^i|\mathbf{a}^{\text{prop}})}{p(\mathbf{a}^i|\mathbf{y})q(\mathbf{a}^{\text{prop}}|\mathbf{a}^i)}\right\} \quad (2)$$

the new state \mathbf{a}^{i+1} is given by \mathbf{a}^{prop} ; otherwise $\mathbf{a}^{i+1} = \mathbf{a}^i$. Under weak conditions of regularity and after a suitable ‘burn-in’, the generated Markov chain provides a dependent sample from the posterior distribution of \mathbf{a} . The question is now how to choose a suitable proposal density q .

A simple and often used proposal density is a multivariate normal density centered at the current value \mathbf{a}^i of the chain and with covariance matrix $h\mathbf{I}$ where h is a user-specified proposal variance and \mathbf{I} is the identity

matrix, i.e. $q(\mathbf{a}^{\text{prop}}|\mathbf{a}^i)$ is the density of $N(\mathbf{a}^i, h\mathbf{I})$. The resulting Metropolis-Hastings algorithm is known as a *random-walk Metropolis* algorithm. In high-dimensional problems, the random-walk Metropolis algorithm may converge very slowly and produce highly auto-correlated samples.

A simple step forward is to use gradient information in the proposal density. The proposal distribution of a *Langevin-Hastings* algorithm [1, 11] is given by

$$N(\mathbf{a}^i + (h/2)\frac{d}{d\mathbf{a}}\log p(\mathbf{a}^i|\mathbf{y}), h\mathbf{I}) \quad (3)$$

where $\frac{d}{d\mathbf{a}}\log p(\mathbf{a}|\mathbf{y})$ is the gradient of the log posterior density - i.e. the vector of derivatives $(d\log p(\mathbf{a}|\mathbf{y})/da_1, \dots, d\log p(\mathbf{a}|\mathbf{y})/da_M)$.

Intuitively, the use of gradient information helps to direct the algorithm towards regions of high posterior density. In applications in spatial statistics [2] the Langevin-Hastings algorithm has proved superior to the random walk Metropolis algorithm. In the context of quantitative genetics, Langevin-Hastings has been successfully applied to implement Bayesian inference in [17], [12], [9] and [3].

When choosing the proposal variance h , rules of thumb suggest that one should aim at acceptance rates of about 25% for random walk and 60% for Langevin-Hastings updates. Single-site schemes where the components in \mathbf{a} are updated in turn may lead to poorly mixing Markov chains due to high correlation between the components of \mathbf{a} .

2.2 Reparameterization

Simulation studies in [7] show that Langevin-Hastings updates may not work well if the components of \mathbf{a} have very different posterior variances. In applications in quantitative genetics, the individuals may contribute with different numbers of observations and may have different numbers of relatives with records. Hence posterior variances may be very different. The correlation structure of the Langevin-Hastings proposal described in the previous section moreover typically differs markedly from the posterior correlation structure where the components are not independent. It may therefore be useful to transform \mathbf{a} into a quantity whose components are less correlated a posteriori. Using the factorisation $\mathbf{A} = \mathbf{T}\mathbf{D}\mathbf{T}^\top$ [8], one may let $\mathbf{a} = \sigma_a\boldsymbol{\gamma}\mathbf{B}^\top$ where $\mathbf{B} = \mathbf{T}\mathbf{D}^{1/2}$ and $\boldsymbol{\gamma}$ is a priori standard normal $N(0, \mathbf{I})$ (note that we regard vectors as row vectors). The posterior correlation matrix of $\boldsymbol{\gamma}$ given \mathbf{y} is then closer to the correlation matrix \mathbf{I} of the Langevin-Hastings proposal. Note that we compute $\mathbf{a} = \sigma_a\boldsymbol{\gamma}\mathbf{B}^\top$ by solving $\mathbf{a}(\mathbf{T}^{-1})^\top = \sigma_a\boldsymbol{\gamma}\mathbf{D}^{1/2}$ with respect to \mathbf{a} . This computation is fast due to the sparseness of \mathbf{T}^{-1} .

The posterior of $\boldsymbol{\gamma}$ given \mathbf{y} is of the form

$$p_{\Gamma}(\boldsymbol{\gamma}|\mathbf{y}) \propto f(\mathbf{y}|\sigma_a\boldsymbol{\gamma}\mathbf{B}^{\top}, \boldsymbol{\beta})p_{\Gamma}(\boldsymbol{\gamma}) \quad (4)$$

where f is the sampling density in (1) and $p_{\Gamma}(\boldsymbol{\gamma})$ denotes the multivariate standard normal density of $\boldsymbol{\gamma}$. Given a current value $\boldsymbol{\gamma}^i$, the Langevin-Hastings proposal is in analogy with (3) of the form

$$\begin{aligned} \boldsymbol{\gamma}^{\text{prop}} &= \boldsymbol{\gamma}^i + (h/2)\frac{\text{d}}{\text{d}\boldsymbol{\gamma}} \log p_{\Gamma}(\boldsymbol{\gamma}^i|\mathbf{y}) + \boldsymbol{\epsilon}^i \\ &= \boldsymbol{\gamma}^i - (h/2)\boldsymbol{\gamma}^i + (h/2)\left[\frac{\text{d}}{\text{d}\mathbf{a}} \log f(\mathbf{y}|\mathbf{a}^i, \boldsymbol{\beta})\right]\sigma_a\mathbf{B} + \boldsymbol{\epsilon}^i \end{aligned} \quad (5)$$

where $-(h/2)\boldsymbol{\gamma}^i = (h/2)\frac{\text{d}}{\text{d}\boldsymbol{\gamma}} \log p_{\Gamma}(\boldsymbol{\gamma})$, $\mathbf{a}^i = \sigma_a\boldsymbol{\gamma}^i\mathbf{B}^{\top}$, $\boldsymbol{\epsilon}^i$ is $N(0, h\mathbf{I})$ distributed, and the chain rule for differentiation is used to obtain the second equation. Letting q_{Γ} denote the corresponding proposal density, the Metropolis-Hastings ratio becomes

$$\frac{f(\mathbf{y}|\sigma_a\boldsymbol{\gamma}^{\text{prop}}\mathbf{B}^{\top}, \boldsymbol{\beta})p_{\Gamma}(\boldsymbol{\gamma}^{\text{prop}})q_{\Gamma}(\boldsymbol{\gamma}^i|\boldsymbol{\gamma}^{\text{prop}})}{f(\mathbf{y}|\sigma_a\boldsymbol{\gamma}^i\mathbf{B}^{\top}, \boldsymbol{\beta})p_{\Gamma}(\boldsymbol{\gamma}^i)q_{\Gamma}(\boldsymbol{\gamma}^{\text{prop}}|\boldsymbol{\gamma}^i)} \quad (6)$$

A posterior sample $\mathbf{a}^1, \mathbf{a}^2, \dots$ is straightforwardly obtained by back-transforming the sample $\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2, \dots$ drawn from (4).

If one prefers to work with the original posterior (1), an equivalent approach to obtain a sample $\mathbf{a}^1, \mathbf{a}^2, \dots$ is to use a proposal \mathbf{a}^{prop} obtained by transforming the proposal (5). More specifically, given the current value \mathbf{a}^i , first compute $\boldsymbol{\gamma}^i = \sigma_a^{-1}\mathbf{a}^i(\mathbf{B}^{\top})^{-1}$, second $\boldsymbol{\gamma}^{\text{prop}}$ using (5), and finally $\mathbf{a}^{\text{prop}} = \sigma_a\boldsymbol{\gamma}^{\text{prop}}\mathbf{B}^{\top}$. The covariance matrix of \mathbf{a}^{prop} then becomes $h\sigma_a^2\mathbf{B}\mathbf{B}^{\top} = h\sigma_a^2\mathbf{A}$, i.e. h times the prior covariance matrix. Moreover, the Metropolis-Hastings ratio in (2) coincides with the ratio (6) since $p(\mathbf{a}|\sigma_a^2) = p_{\Gamma}(\boldsymbol{\gamma})/|\sigma_a\mathbf{B}^{\top}|$, $q(\mathbf{a}^{\text{prop}}|\mathbf{a}^i) = q_{\Gamma}(\boldsymbol{\gamma}^{\text{prop}}|\boldsymbol{\gamma}^i)/|\sigma_a\mathbf{B}^{\top}|$, and $q(\mathbf{a}^i|\mathbf{a}^{\text{prop}}) = q_{\Gamma}(\boldsymbol{\gamma}^i|\boldsymbol{\gamma}^{\text{prop}})/|\sigma_a\mathbf{B}^{\top}|$ where $|\sigma_a\mathbf{B}^{\top}|$ is the Jacobian for the transformation from $\boldsymbol{\gamma}$ to \mathbf{a} . A more general perspective on the use of reparameterization is given in Appendix A.

2.3 Normal approximation of the posterior

Suppose for a moment that $q(\mathbf{a}^{\text{prop}}|\mathbf{a}^i)$ is equal to the target density $p(\mathbf{a}^{\text{prop}}|\mathbf{y})$. The Metropolis-Hastings algorithm then produces independent draws from the posterior. This indicates that an efficient Metropolis-Hastings algorithm might be obtained by constructing a proposal density which is a good approximation of the posterior density

Consider the second-order Taylor expansion

$$\begin{aligned} \log p(\mathbf{a}^{\text{prop}}|\mathbf{y}) &\approx \\ \log p(\hat{\mathbf{a}}|\mathbf{y}) + (\mathbf{a}^{\text{prop}} - \hat{\mathbf{a}}) \frac{d}{d\mathbf{a}} \log p(\hat{\mathbf{a}}|\mathbf{y})^\top - \frac{1}{2}(\mathbf{a}^{\text{prop}} - \hat{\mathbf{a}})\mathbf{H}(\hat{\mathbf{a}})(\mathbf{a}^{\text{prop}} - \hat{\mathbf{a}})^\top \end{aligned} \quad (7)$$

around a value $\hat{\mathbf{a}}$ where $\mathbf{H}(\mathbf{a}) = -\frac{d^2}{d\mathbf{a}^\top d\mathbf{a}} \log p(\mathbf{a}|\mathbf{y}) = \mathbf{A}^{-1}/\sigma_a^2 - \frac{d^2}{d\mathbf{a}^\top d\mathbf{a}} \log f(\mathbf{y}|\mathbf{a})$ is minus the Hessian matrix of second derivatives. Suppose $\mathbf{H}(\hat{\mathbf{a}})$ is positive definite. By the identity

$$\begin{aligned} &-\frac{1}{2}(\mathbf{x} - \mathbf{m} - \mathbf{b}\mathbf{C}^{-1})\mathbf{C}(\mathbf{x} - \mathbf{m} - \mathbf{b}\mathbf{C}^{-1})^\top \\ &= -\frac{1}{2}\mathbf{b}\mathbf{C}^{-1}\mathbf{b}^\top + (\mathbf{x} - \mathbf{m})\mathbf{b}^\top - \frac{1}{2}(\mathbf{x} - \mathbf{m})\mathbf{C}(\mathbf{x} - \mathbf{m})^\top \end{aligned}$$

(for appropriate row vectors \mathbf{x} , \mathbf{m} , \mathbf{b} and a positive definite matrix \mathbf{C}), the exponential of the right hand side of (7) is proportional to the density

$$q(\mathbf{a}^{\text{prop}}|\mathbf{a}^i) \propto |\mathbf{H}(\hat{\mathbf{a}})|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{a}^{\text{prop}} - \mu(\hat{\mathbf{a}}))\mathbf{H}(\hat{\mathbf{a}})(\mathbf{a}^{\text{prop}} - \mu(\hat{\mathbf{a}}))^\top\right) \quad (8)$$

of a multivariate normal distribution with mean $\mu(\hat{\mathbf{a}}) = \hat{\mathbf{a}} + \frac{d}{d\mathbf{a}} \log p(\hat{\mathbf{a}}|\mathbf{y})\mathbf{H}(\hat{\mathbf{a}})^{-1}$ and precision matrix $\mathbf{H}(\hat{\mathbf{a}})$.

Several options are available for choosing $\hat{\mathbf{a}}$. One may e.g. let $\hat{\mathbf{a}}$ be given by the current value \mathbf{a}^i in the Markov chain or by the result of one Newton-Raphson step starting from \mathbf{a}^i . When $\hat{\mathbf{a}}$ depends on \mathbf{a}^i then so does the value $\hat{\mathbf{a}}^{\text{prop}}$, say, used when evaluating the proposal density

$$q(\mathbf{a}^i|\mathbf{a}^{\text{prop}}) \propto |\mathbf{H}(\hat{\mathbf{a}}^{\text{prop}})|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{a}^i - \mu(\hat{\mathbf{a}}^{\text{prop}}))\mathbf{H}(\hat{\mathbf{a}}^{\text{prop}})(\mathbf{a}^i - \mu(\hat{\mathbf{a}}^{\text{prop}}))^\top\right).$$

Thus in the Metropolis-Hastings ratio (2) we need to evaluate the ratio of determinants $|\mathbf{H}(\hat{\mathbf{a}}^{\text{prop}})|^{1/2}/|\mathbf{H}(\hat{\mathbf{a}})|^{1/2}$ using e.g. sparse matrix methods (see Section 2.4). Alternatively one may iterate Newton-Raphson to convergence so that $\hat{\mathbf{a}}$ becomes the mode of the posterior which does not depend on \mathbf{a}^i . Then $\mu(\hat{\mathbf{a}}) = \hat{\mathbf{a}}$ and the ratio of determinants conveniently becomes one.

Sampling from a normal approximation is discussed in Section 2.4.

2.4 Implementation of the normal approximation

Typically, the genetic random effects enter the sampling density $f(\mathbf{y}|\mathbf{a})$ via a linear predictor $\boldsymbol{\eta} = \mathbf{a}\mathbf{Z}$ (i.e. $f(\mathbf{y}|\mathbf{a}) = \tilde{f}(\mathbf{y}|\boldsymbol{\eta})$) where \mathbf{Z} is an incidence matrix

relating the observed traits to the random effects. The precision matrix $\mathbf{H}(\hat{\mathbf{a}})$ in the normal approximation proposal density (8) then takes the form

$$\mathbf{A}^{-1}/\sigma_a^2 + \mathbf{Z}\boldsymbol{\Sigma}^{-1}\mathbf{Z}^\top$$

where $\boldsymbol{\Sigma}^{-1} = -\frac{d^2}{d\boldsymbol{\eta}^\top d\boldsymbol{\eta}} \log \tilde{f}(\mathbf{y}|\boldsymbol{\eta})|_{\boldsymbol{\eta}=\hat{\mathbf{a}}\mathbf{Z}}$. The normal approximation proposal distribution is thus formally equivalent to the conditional distribution of random effects in a ‘virtual’ linear normal model

$$\tilde{\mathbf{y}} = \mathbf{a}^{\text{prop}}\mathbf{Z} + \tilde{\boldsymbol{\epsilon}}$$

where \mathbf{a}^{prop} is $N(0, \sigma_a^2\mathbf{A})$, $E[\mathbf{a}^{\text{prop}}|\tilde{\mathbf{y}}] = \mu(\hat{\mathbf{a}})$, and $\tilde{\boldsymbol{\epsilon}} \sim N(0, \boldsymbol{\Sigma})$ and $\tilde{\mathbf{y}}$ represents ‘virtual’ noise and data. Hence we may sample from the normal approximation by applying the García-Cortés and Sorensen [4] algorithm based on the decomposition

$$\mathbf{a}^{\text{prop}} = (\mathbf{a}^{\text{prop}} - E[\mathbf{a}^{\text{prop}}|\tilde{\mathbf{y}}]) + E[\mathbf{a}^{\text{prop}}|\tilde{\mathbf{y}}] = \mathbf{e} + \mu(\hat{\mathbf{a}})$$

where the ‘prediction error’ $\mathbf{e} = (\mathbf{a}^{\text{prop}} - E[\mathbf{a}^{\text{prop}}|\tilde{\mathbf{y}}])$ and the ‘prediction’ $\mu(\hat{\mathbf{a}}) = E[\mathbf{a}^{\text{prop}}|\tilde{\mathbf{y}}]$ are independent. Hence if \mathbf{e}_{sim} is a simulation of \mathbf{e} then

$$\mathbf{a}_{\text{sim}} = \mathbf{e}_{\text{sim}} + \mu(\hat{\mathbf{a}})$$

is a conditional simulation of \mathbf{a}^{prop} given $\tilde{\mathbf{y}}$. The simulated prediction error \mathbf{e}_{sim} may be generated as follows

1. simulate $(\mathbf{a}_{\text{sim}}, \tilde{\mathbf{y}}_{\text{sim}})$ from the joint distribution of $(\mathbf{a}^{\text{prop}}, \tilde{\mathbf{y}})$ (using the Henderson factorization $\mathbf{A} = \mathbf{T}\mathbf{D}\mathbf{T}^\top$)
2. compute $\mu(\hat{\mathbf{a}}_{\text{sim}}) = E[\mathbf{a}^{\text{prop}}|\tilde{\mathbf{y}}_{\text{sim}}]$ by solving the standard mixed model equations $\mu(\hat{\mathbf{a}}_{\text{sim}})[\mathbf{A}^{-1}/\sigma_a^2 + \mathbf{Z}\boldsymbol{\Sigma}^{-1}\mathbf{Z}^\top] = \mathbf{y}_{\text{sim}}\boldsymbol{\Sigma}^{-1}\mathbf{Z}^\top$ for the virtual linear model
3. return $\mathbf{e}_{\text{sim}} = \mathbf{a}_{\text{sim}} - \mu(\hat{\mathbf{a}}_{\text{sim}})$.

Alternatively, one may exploit the sparseness of $\mathbf{H}(\hat{\mathbf{a}})$ which enables fast computation of the Cholesky factorization of $\mathbf{H}(\hat{\mathbf{a}})$, (see [13] and [14]). For the latter approach the `c` library `GMRFLib` (www.math.ntnu.no/~hrue/GMRFLib/) and Appendix B in Rue and Knorr-Held, 2005) provides an extensive suite of procedures for computation of and sampling from normal approximations. Using this library, sophisticated MCMC algorithms can be constructed with little programming effort. Note that with sparse Cholesky factorizations, it is straightforward to evaluate ratios of determinants $|\mathbf{H}(\hat{\mathbf{a}}^{\text{prop}})|^{1/2}/|\mathbf{H}(\hat{\mathbf{a}})|^{1/2}$ possibly appearing in the Metropolis-Hastings ratio (2). `GMRFLib` is used in [18] to implement Bayesian inference for a multiple trait model.

2.5 Updating σ_a^2

Introducing updates of the parameters σ_a^2 and $\boldsymbol{\beta}$ is in principle straightforward but regarding σ_a^2 there are some issues concerning whether we consider the original random effects \mathbf{a} or reparameterized random effects $\boldsymbol{\gamma}$, cf. Section 2.2.

Let $p(\sigma_a^2)$ denote the prior for σ_a^2 and suppose that given a current value $\sigma_a^{2,i}$, we generate a proposal $\sigma_a^{2,\text{prop}}$ from some proposal density q_Σ . Depending on whether we use the original random effects or not, the Metropolis-Hastings ratios become

$$\frac{p(\mathbf{a}^i | \sigma_a^{2,\text{prop}}) p(\sigma_a^{2,\text{prop}}) q_\Sigma(\sigma_a^{2,i} | \sigma_a^{2,\text{prop}})}{p(\mathbf{a}^i | \sigma_a^{2,i}) p(\sigma_a^{2,i}) q_\Sigma(\sigma_a^{2,\text{prop}} | \sigma_a^{2,i})} \quad (9)$$

or

$$\frac{f(\mathbf{y} | \sigma_a^{\text{prop}} \boldsymbol{\gamma}^i \mathbf{B}^\top, \boldsymbol{\beta}) p(\sigma_a^{2,\text{prop}}) q_\Sigma(\sigma_a^{2,i} | \sigma_a^{2,\text{prop}})}{f(\mathbf{y} | \sigma_a^{2,i} \boldsymbol{\gamma}^i \mathbf{B}^\top, \boldsymbol{\beta}) p(\sigma_a^{2,i}) q_\Sigma(\sigma_a^{2,\text{prop}} | \sigma_a^{2,i})} \quad (10)$$

which do not coincide. In particular, note that updating σ_a^2 with reparameterized random effects effectively updates \mathbf{a} too with $\mathbf{a}^{i+1} = \sigma_a^{i+1} \boldsymbol{\gamma}^i \mathbf{B}^\top$.

This suggests a joint update of (σ_a^2, \mathbf{a}) in the case of no reparameterization. If $(\sigma_a^{2,i}, \mathbf{a}^i)$ denotes the current joint value, the proposal is $(\sigma_a^{2,\text{prop}}, \mathbf{a}^{\text{prop}})$ where $\sigma_a^{2,\text{prop}}$ is generated from q_Σ and $\mathbf{a}^{\text{prop}} = \sigma_a^{\text{prop}} \mathbf{a}^i / \sigma_a^i = \sigma_a^{\text{prop}} \boldsymbol{\gamma}^i \mathbf{B}^\top$ where $\boldsymbol{\gamma}^i = (\sigma_a^i)^{-1} \mathbf{a}^i (\mathbf{B}^{-1})^\top$. Note that the $M+1$ dimensional proposal is generated using a one-dimensional proposal density. Hence we need a slightly more general version of the Metropolis-Hastings algorithm reviewed in [19] and in [16]. The Metropolis-Hastings ratio becomes

$$\frac{f(\mathbf{y} | \mathbf{a}^{\text{prop}}, \boldsymbol{\beta}) p(\mathbf{a}^{\text{prop}} | \sigma_a^{2,\text{prop}}) p(\sigma_a^{2,\text{prop}}) q_\Sigma(\sigma_a^{2,i} | \sigma_a^{2,\text{prop}})}{f(\mathbf{y} | \mathbf{a}^i, \boldsymbol{\beta}) p(\mathbf{a}^i | \sigma_a^{2,i}) p(\sigma_a^{2,i}) q_\Sigma(\sigma_a^{2,\text{prop}} | \sigma_a^{2,i})} \left(\frac{\sigma_a^{\text{prop}}}{\sigma_a^i} \right)^M \quad (11)$$

where $(\sigma_a^{\text{prop}} / \sigma_a^i)^M$ is the Jacobian of the transformation

$$g(\sigma_a^{2,i}, \mathbf{a}^i, \sigma_a^{2,\text{prop}}) = (\sigma_a^{2,\text{prop}}, (\sigma_a^{\text{prop}} / \sigma_a^i) \mathbf{a}^i, \sigma_a^{2,i}),$$

see Section 4 in [19] or Section 11.7 in [16] for details. Note that (11) coincides with (10) since $(\sigma_a^{\text{prop}})^M p(\mathbf{a}^{\text{prop}} | \sigma_a^{2,\text{prop}}) = (\sigma_a^i)^M p(\mathbf{a}^i | \sigma_a^{2,i})$. A more general perspective is given in Appendix A.

If one restricts attention to a χ^{-2} prior and the case without reparameterization of the genetic effects, then the full conditional

$$p(\sigma_a^2 | \mathbf{y}, \mathbf{a}) \propto p(\mathbf{a} | \sigma_a^2) p(\sigma_a^2) \quad (12)$$

becomes χ^{-2} too. Hence a Gibbs update might be used in which case we sample the new value directly from the full conditional. Note, however,

that in cases of \mathbf{a} of high dimension M , the full conditional (12) becomes highly concentrated around the maximum likelihood estimate $\mathbf{a}A^{-1}\mathbf{a}^\top/M = \arg \max_{\sigma_a^2} p(\mathbf{a}|\sigma_a^2)$ given \mathbf{a} . Hence, \mathbf{a} and σ_a^2 are strongly correlated a posteriori and it is our experience that it is advantageous to use instead the approach with reparameterized random effects/joint update of (\mathbf{a}, σ_a^2) .

In our data examples we could apply a 3-5 times larger proposal standard deviation in q_Σ when used in combination with (10) while maintaining the same acceptance rates as for the case without reparameterization (9). More discussion on joint updates of random effects and variance parameters can be found in [10].

2.6 Comparison of samplers in terms of Monte Carlo error and computational cost

Given an MCMC sample $(\mathbf{a}^1, \theta^1), (\mathbf{a}^2, \theta^2), \dots, (\mathbf{a}^n, \theta^n)$ from the posterior distribution of (\mathbf{a}, θ) , $\theta = (\boldsymbol{\beta}, \sigma_a^2)$, and some function $h(\mathbf{a}, \theta)$, the posterior expectation

$$\mathbb{E}[h(\mathbf{a}, \theta)|\mathbf{y}] = \int h(\mathbf{a}, \theta)p(\mathbf{a}, \theta|\mathbf{y})d\mathbf{a}d\theta$$

is estimated by the average $\bar{h}_n = \sum_{i=1}^n h(\mathbf{a}^i, \theta^i)/n$. The Monte Carlo variance of \bar{h}_n is given by V_{asymp}/n , where

$$V_{\text{asymp}} = \lim_{n \rightarrow \infty} \text{Var} \sqrt{n}\bar{h}_n = \text{Var}[h(\mathbf{a}, \theta)|y](1 + 2 \sum_{m=1}^{\infty} \rho_m)$$

is the so-called asymptotic variance given in terms of the posterior variance $\text{Var}[h(\mathbf{a}, \theta)|y]$ and the Markov chain lag- m autocorrelations

$$\rho_m = \text{Corr}[h(\mathbf{a}^k, \theta^k), h(\mathbf{a}^{k+m}, \theta^{k+m})].$$

To attain a given Monte Carlo variance of size V , we need a sample size of $n_V = V_{\text{asymp}}/V$ and if the cost of generating one sample is c then the total cost becomes $cn_V = c\tau \text{Var}[h(\mathbf{a}, \theta)|y]/V$ where $\tau = 1 + 2 \sum_{m=1}^{\infty} \rho_m$ is the integrated autocorrelation. Thus $c\tau$ is an appropriate performance measure for an MCMC algorithm. The so-called effective sample size is given by n/τ . The integrated autocorrelation can be estimated as suggested e.g. in [5].

Note that the ratio τ_2/τ_1 of integrated autocorrelations for two MCMC samplers is equal to the ratio n_2/n_1 of numbers of iterations n_2 and n_1 required to obtain the same MCMC variance V with the two samplers.

3 A model with genetically structured variance heterogeneity

We now discuss the methods of the previous sections within the context of the [15] model for genetically structured variance heterogeneity. For ease of presentation systematic and environmental effects are omitted. Let \mathbf{a}^* denote random effects affecting the residual variance of \mathbf{y} . Given \mathbf{a} and \mathbf{a}^* , the components y_i of \mathbf{y} are independent $N(\mu + \mathbf{a}\mathbf{z}_i^\top, \exp(\mu^* + \mathbf{a}^*\mathbf{z}_i^\top))$ where \mathbf{z}_i is an incidence vector with components equal to zero or one. Hence the same type of linear model is applied for the mean and the log residual variance of y_i where the mean depends on \mathbf{a} and the log residual variance on \mathbf{a}^* . The joint distribution of \mathbf{a} and \mathbf{a}^* is zero mean normal with covariance matrix $\mathbf{G} \otimes \mathbf{A}$ where

$$\mathbf{G} = \begin{bmatrix} \sigma_a^2 & \rho\sigma_a\sigma_{a^*} \\ \rho\sigma_a\sigma_{a^*} & \sigma_{a^*}^2 \end{bmatrix}.$$

The correlation between the two types of random effects is given by ρ , and σ_a^2 and $\sigma_{a^*}^2$ are the variances for the genetic random effects.

3.1 Reparameterizations

Various reparameterization strategies are possible for this model. Letting $\mathbf{U}_{\mathbf{G}}$ denote the upper triangular Cholesky factor of \mathbf{G} (i.e. $\mathbf{G} = \mathbf{U}_{\mathbf{G}}^\top \mathbf{U}_{\mathbf{G}}$), $(\mathbf{a}, \mathbf{a}^*) = (\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)\mathbf{U}_{\mathbf{G}} \otimes \mathbf{B}^\top$ where $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ is a standard normal vector and \mathbf{B} is defined in Section 2.2. Langevin-Hastings updates for the vector of a priori uncorrelated random effects $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ are used in [17], [12], and [9]. An alternative reparameterization is based on $(\mathbf{a}, \mathbf{a}^*) = (\mathbf{a}, \alpha\mathbf{a} + \mathbf{u})$ where $\alpha = \rho\sigma_{a^*}/\sigma_a$ and $\mathbf{u} = \mathbf{a}^* - \mathbf{E}[\mathbf{a}^*|\mathbf{a}]$. Then \mathbf{u} is $N(0, \sigma_a^2(1 - \rho^2)\mathbf{A})$ and a priori independent of \mathbf{a} . Hence one might update \mathbf{a} and \mathbf{u} in turn hoping that these quantities are only weakly correlated a posteriori. Note that it is not guaranteed that the Hessian matrix with respect to \mathbf{u} is positive definite.

3.2 Normal approximations

Let $\boldsymbol{\Sigma} = \text{diag}(\sigma_i^2)$ and $\mathbf{R} = \text{diag}(r_i)$ denote diagonal matrices where $\sigma_i^2 = \exp(\mu^* + \mathbf{a}^*\mathbf{z}_i^\top)$ is the conditional variance given \mathbf{a}^* for the i th observation and $r_i = (y_i - \mu - \mathbf{a}\mathbf{z}_i^\top)$ is the i th residual. The precision matrix in the normal approximation of the posterior for $(\mathbf{a}, \mathbf{a}^*)$ is then

$$\begin{bmatrix} \mathbf{Z}\boldsymbol{\Sigma}^{-1}\mathbf{Z}^\top + \mathbf{A}^{-1}g^{11} & \mathbf{Z}\boldsymbol{\Sigma}^{-1}\mathbf{W}^\top + \mathbf{A}^{-1}g^{12} \\ \mathbf{W}\boldsymbol{\Sigma}^{-1}\mathbf{Z}^\top + \mathbf{A}^{-1}g^{12} & \frac{1}{2}\mathbf{W}\boldsymbol{\Sigma}^{-1}\mathbf{W}^\top + \mathbf{A}^{-1}g^{22} \end{bmatrix} \quad (13)$$

where $\mathbf{W} = \mathbf{Z}\mathbf{R}$ and g^{ij} are the entries of \mathbf{G}^{-1} . Due to the factor $1/2$ in the lower right block, this matrix cannot be recognized as the covariance matrix of a conditional normal distribution and there is in fact no guarantee that it is positive definite. This is further illustrated in the toy example in Section 3.3 which shows that the joint posterior of $(\mathbf{a}, \mathbf{a}^*)$ can be far from multivariate normal. In Section 4 we use the normal approximation in turn for \mathbf{a} and \mathbf{a}^* separately. In this case the covariance matrices are given by the diagonal blocks in (13).

3.3 Toy example

We illustrate the various reparameterization strategies in the very simple case where $\mathbf{y} = (-2.62, -2.42)$ consists of two observations, and \mathbf{a} and \mathbf{a}^* are one-dimensional. Figure 1 shows the posterior densities of $(\mathbf{a}, \mathbf{a}^*)$, $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ and (\mathbf{a}, \mathbf{u}) in the case where $\mu = 0$, $\mu^* = -1$, $\sigma_a^2 = 1$, $\sigma_{a^*}^2 = 0.25$, and $\rho = 0.75$. The plots demonstrate for the given parameter settings that $(\mathbf{a}, \mathbf{a}^*)$ are highly correlated a posteriori and that the joint posterior distribution of $(\mathbf{a}, \mathbf{a}^*)$ is not well approximated by a normal distribution. The transformed random effects $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}^*$ seem approximately uncorrelated but have different posterior variances. As expected, \mathbf{a} and \mathbf{u} are less correlated a posteriori than \mathbf{a} and \mathbf{a}^* but the joint distribution is far from normal.

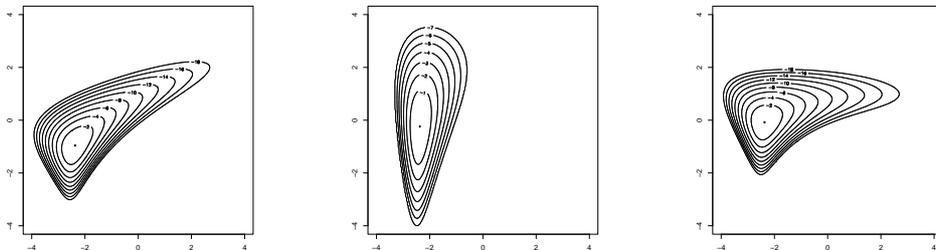


Figure 1: Posterior densities for $(\mathbf{a}, \mathbf{a}^*)$ (left), $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ (middle), and (\mathbf{a}, \mathbf{u}) (right). Note that the modal value of the posterior density is subtracted in all four plots and that contour curves are omitted for very low values of the posterior densities.

The plots in Figure 2 illustrate the random walk and Langevin-Hastings proposals. The Langevin-Hastings proposal mean lies in a region of higher posterior density than the current value. This means that proposals in a rather large region around the proposal mean have a good chance of being accepted. Figure 3 shows that the normal approximation is poor for the joint

posterior of $(\mathbf{a}, \mathbf{a}^*)$ while it works well for the conditional posteriors of $\mathbf{a}|\mathbf{y}, \mathbf{a}^*$ and $\mathbf{a}^*|\mathbf{y}, \mathbf{a}$ separately.

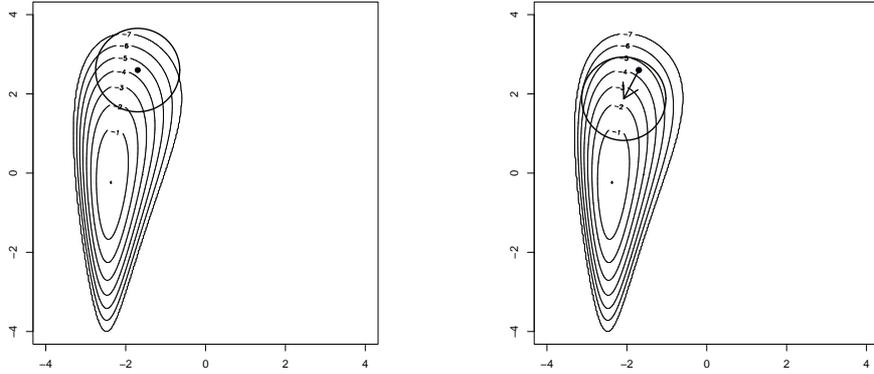


Figure 2: Illustration of random walk and Langevin-Hastings for (γ, γ^*) with $h = 0.4$. In both plots, the bold dot represents a current state \mathbf{a}^i and 75 % of the proposals fall within the circle. In the right plot the proposal mean is given by the current value plus $h \frac{d}{d\mathbf{a}} \log p(\mathbf{a}^i|\mathbf{y})/2$ indicated by the arrow.

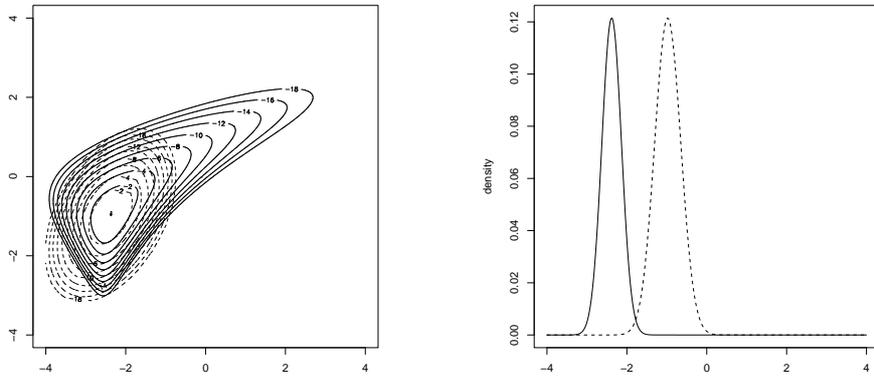


Figure 3: Illustration of normal approximation (dashed contours) evaluated at the mode for the joint posterior (solid contours) of $(\mathbf{a}, \mathbf{a}^*)$ (left) and for the conditional distributions of $\mathbf{a}|\mathbf{a}^*, \mathbf{y}$ (solid curve) and $\mathbf{a}^*|\mathbf{a}, \mathbf{y}$ (dashed curve).

4 Examples

In this section we compare the performance of Langevin-Hastings and normal approximation MCMC algorithms applied to three data sets which have been previously analyzed in [17], [12], and [9]. The first data set originates from a selection experiment for pig litter size and contains 10,060 litter size records from 4,149 sows. The pedigree file includes 6,437 individuals. The second data set contains 2,996 litter sizes from a divergent selection experiment for rabbit uterine capacity with 1,161 individuals in the pedigree. The third and largest data set consists of weights for each of 22,033 adult snails and the pedigree file includes 22,454 individuals. For all three datasets we consider the model from Section 3 extended with systematic and environmental effects and impose prior distributions on the unknown location and variance parameters. The posterior means of the genetic covariance parameters are given in the last three columns of Table II. The genetic correlation parameter ρ is negative for the pig and rabbit litter size data while it is positive for the snail weights. More details on the data, priors, and posterior results can be found in [17], [12] and [9].

The first algorithm LH is the one employed in [17], [12] and [9] where reparameterized random effects $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ are considered and the components of the posterior distribution are updated in turn using Langevin-Hastings updates for $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ and either Langevin-Hastings or random walk updates for the other components (for the positive variance parameters we apply the random walk updates to the log variances). The second algorithm NX is as LH except that the Langevin-Hastings update for $(\boldsymbol{\gamma}, \boldsymbol{\gamma}^*)$ is replaced with normal approximation updates for \mathbf{a} and \mathbf{a}^* separately. Following Section 2.5, for NX we use a joint update of the genetic random effects and the variance parameters in order to obtain an update that is equivalent to the update of the variance parameters for the LH algorithm. The normal approximation updates are implemented using `GMRFLib`. The proposal variances for the Langevin-Hastings and random walk updates are chosen according to the rules of thumb mentioned in Section 2.1.

Table I shows ratios of integrated autocorrelations τ obtained using respectively LH and NX. The integrated autocorrelations are evaluated for quadratic forms involving \mathbf{a} and \mathbf{a}^* and the first two components of \mathbf{a} and \mathbf{a}^* . Considering e.g. the random effect a_1 for the pigs data, the integrated autocorrelation is 873 times larger for LH than for NX. This means that 873 times more iterations are needed with LH to obtain the same precision as for the NX algorithm (see Section 2.6 for details concerning integrated autocorrelation and MCMC precision). Columns 2-4 in Table II show ratios of integrated autocorrelations for the genetic covariance parameters using

respectively LH and NX. Regarding integrated autocorrelation, NX clearly outperforms LH.

Data	$\mathbf{a}\mathbf{A}^{-1}\mathbf{a}^\top$	$\mathbf{a}\mathbf{A}^{-1}\mathbf{a}^{*\top}$	$\mathbf{a}^*\mathbf{A}^{-1}\mathbf{a}^{*\top}$	a_1	\tilde{a}_1
Rabbits	54	59	61	105	106
Pigs	315	359	280	873	673
Snails	317	393	166	465	253

Table I: Ratios of integrated autocorrelations (LH/NX) for quadratic forms and two genetic random effects using LH and NX

To evaluate the performance of the algorithms, computing time c must be taken into account. Column 5 in Table II shows that the computing time for one MCMC iteration is between 20 to 100 times higher for the NX algorithm than for the LH algorithm. The ratios of the products τc for respectively LH and NX are between 3-6 for the rabbits, 1-9 for the pigs, and 5-13 for the snails depending on the parameters considered. Hence, NX is superior for all datasets.

Data	σ_a^2	$\sigma_{a^*}^2$	ρ	c	$E[\sigma_a^2 \mathbf{y}]$	$E[\sigma_{a^*}^2 \mathbf{y}]$	$E[\rho \mathbf{y}]$
Rabbits	102	81	117	20	0.82	0.16	-0.74
Pigs	129	190	278	100	1.62	0.10	-0.62
Snails	328	158	401	35	1.71	0.29	0.81

Table II: Column 2-4: Ratios of integrated autocorrelations (LH/NX) for genetic variance parameters using LH and NX. Column c contains the ratios of computing times (NX/LH) for one MCMC iteration. Last three columns contain posterior means of the genetic covariance parameters

For the data set of snail weights with high-dimensional \mathbf{a}^* , the acceptance rate for the normal approximation updates of \mathbf{a}^* is quite small – around 5% – while larger acceptance rates 50% and 30 % are obtained for the smaller rabbits and pigs data sets. If the MCMC algorithm is initialized in values far from the posterior mode, the acceptance probability for the normal approximation update of \mathbf{a}^* may be very small in which case a large burn-in is needed.

The long computing times for NX and the large integrated autocorrelations for LH suggest to consider an algorithm where one alternates between normal approximation and Langevin-Hastings updates. More specifically we obtain an algorithm NXLH by replacing the normal approximation update by a Langevin-Hastings update in every second iteration. The Langevin-Hastings update is obtained via a Langevin-Hastings update for the transformed genetic effects as described in the end of Section 2.2. We considered

NXLH for the rabbits data and compared with NX the computing cost is roughly halved while the integrated autocorrelations only slightly increase. The ratios of τc for LH and NXLH are between 4 and 9 so that NXLH in fact works better than both NX and LH. This algorithm may also be advantageous in cases where NX requires a long burn-in.

We also tried out normal approximations for the (\mathbf{a}, \mathbf{u}) reparameterization but this did not offer any improvement.

5 Discussion

Normal approximation proposal distributions are intuitively appealing and provide smaller integrated autocorrelations than Langevin-Hastings updates in the examples in Section 4. A distinct advantage of the normal approximation updates is moreover that they do not require user tuning of proposal variances. The computing time for the normal approximation updates is high but sensitive to the choice of implementation. The normal approximation updates used in Section 4 are implemented using general routines in `GMRFLib` based on numerical methods for sparse matrices. This approach reduces very much the programming effort but one loses the computational advantages offered by the specific structure of the genetic correlation matrix. For the rabbits data, [9] reduces the computing cost of the NX algorithm by a factor three using the approach described in Section 2.4 where samples from the normal approximation are obtained using the García-Cortés and Sorensen [4] algorithm and the Henderson factorization.

Finally, using a mixture of Langevin-Hastings and normal approximation updates may work even better than pure Langevin-Hastings or normal approximation algorithms since the use of Langevin-Hastings saves computing time while the normal approximation maintains small integrated autocorrelations. This option is also helpful in cases where a pure normal approximation algorithm requires a long burn-in.

Acknowledgement We thank two anonymous referees for helpful comments on a previous version of the manuscript.

References

- [1] BESAG, J. Discussion on the paper by Grenander and Miller. *J. Roy. Statist. Soc. Ser. B* 56 (1994), 591–592.

- [2] CHRISTENSEN, O. F., AND WAAGEPETERSEN, R. P. Bayesian prediction of spatial count data using generalized linear mixed models. *Biometrics* 58 (2002), 280–286.
- [3] DAMGAARD, L. H. Joint quantitative genetic analysis of survival, linear Gaussian and ordered categorical traits. In *Proceedings of the 8th World Congress of Genetics Applied to Livestock Production* (Belo Horizonte, Minas Gerais, Brazil, August 13-18 2006).
- [4] GARCÍA-CORTÉS, L. A., AND SORENSEN, D. On a multivariate implementation of the Gibbs sampler. *Genet. Sel. Evol.* 28 (1996), 121–126.
- [5] GEYER, C. J. Practical Markov chain Monte Carlo. *Statist. Sci.* 7 (1992), 473–511.
- [6] GEYER, C. J., AND THOMPSON, E. A. Constrained Monte Carlo maximum likelihood for dependent data. *J. Roy. Soc. Statist. Ser. B* 54 (1992), 657–699.
- [7] GUSTAFSON, P., MACNAB, Y. C., AND WEN, S. On the value of derivative evaluations and random walk suppression in Markov chain Monte Carlo algorithms. *Stat. Comput.* 14 (2004), 23–38.
- [8] HENDERSON, C. R. A simple method for the inverse of a numerator relationship matrix used in prediction of breeding values. *Biometrics* 32 (1976), 69–83.
- [9] IBÁÑEZ-ESCRICHE, N. *Models for residual variances in quantitative genetics. An application to rabbit uterine capacity*. PhD thesis, Departamento de Ciencia Animal, Universidad Politécnica de Valencia, Valencia, 2006.
- [10] KNORR-HELD, L., AND RUE, H. On block updating in Markov random field models for disease mapping. *Scand. J. Statist.* 29 (2002), 597–614.
- [11] ROBERTS, G. O., AND TWEEDIE, R. L. Exponential convergence of Langevin diffusions and their discrete approximations. *Bernoulli* 2 (1996), 341–363.
- [12] ROS, M., SORENSEN, D., WAAGEPETERSEN, R., DUPONT-NIVET, M., SANCRISTOBAL, M., BONNET, J.-C., AND MALLARD, J. Evidence for genetic control of adult weight plasticity in the snail *Helix aspersa*. *Genetics* 168 (2004), 2089–2097.

- [13] RUE, H. Fast sampling of Gaussian Markov random fields. *J. Roy. Statist. Soc. Ser. B* 63 (2001), 325–338.
- [14] RUE, H., AND KNORR-HELD, L. *Gaussian Markov random fields - theory and applications*. Chapman & Hall/CRC, 2005.
- [15] SAN CRISTOBAL-GAUDY, M., ELSEN, J. M., BODIN, L., AND CHEVALET, C. Prediction of the response to a selection for canalisation of a continuous trait in animal breeding. *Genet. Sel. Evol.* 30 (1998), 423–451.
- [16] SORENSEN, D., AND GIANOLA, D. *Likelihood, Bayesian, and Markov chain Monte Carlo methods in quantitative genetics*. Springer-Verlag, 2002.
- [17] SORENSEN, D., AND WAAGEPETERSEN, R. Normal linear models with genetically structured variance heterogeneity: a case study. *Genet. Res.* 82 (2003), 207–222.
- [18] STEINSLAND, I., AND JENSEN, H. Making inference from Bayesian animal models utilising Gaussian Markov random field properties. Statistics Preprint 10, Norwegian University of Science and Technology, 2005.
- [19] WAAGEPETERSEN, R., AND SORENSEN, D. A tutorial on reversible jump MCMC with a view towards applications in QTL-mapping. *International Statistical Review* 69 (2001), 49–61.

A Reparameterization and reversibility

Let \mathbf{X} denote a random vector with density f and let $\tilde{\mathbf{X}} = g(\mathbf{X})$ where g is a one-to-one mapping with inverse g^{-1} . Then $\tilde{\mathbf{X}}$ has density $\tilde{f}(\tilde{\mathbf{x}}) = f(\mathbf{x})/|g'(\mathbf{x})|$ where $\mathbf{x} = g^{-1}(\tilde{\mathbf{x}})$. Suppose we generate a Metropolis-Hastings chain $\mathbf{X}^1, \mathbf{X}^2, \dots$ in order to sample f . The key point of the Metropolis-Hastings update is reversibility, ie. $(\mathbf{X}^i, \mathbf{X}^{i+1})$ has the same distribution as $(\mathbf{X}^{i+1}, \mathbf{X}^i)$ provided \mathbf{X}^i is distributed according to f . On the other hand, if \mathbf{X}^i is distributed according to f then $\tilde{\mathbf{X}}^i = g(\mathbf{X}^i)$ is distributed according to \tilde{f} . Hence, if we apply a Metropolis-Hastings update to $\tilde{\mathbf{X}}^i$ and obtain $\tilde{\mathbf{X}}^{i+1}$, then we have reversibility for the pair $(\tilde{\mathbf{X}}^i, \tilde{\mathbf{X}}^{i+1})$. Thus, if we backtransform to obtain $\mathbf{X}^{i+1} = g^{-1}(\tilde{\mathbf{X}}^{i+1})$ then we have also have reversibility for $(\mathbf{X}^i, \mathbf{X}^{i+1})$. To summarize, given \mathbf{X}^i , we can either update \mathbf{X}^i directly or instead transform, update $\tilde{\mathbf{X}}^i$, and backtransform. Suppose we update $\tilde{\mathbf{X}}^i$

using a proposal $\tilde{\mathbf{X}}^{\text{prop}}$ generated from a proposal density \tilde{q} . This is then in fact equivalent to updating \mathbf{X}^i using the proposal $\mathbf{X}^{\text{prop}} = g^{-1}(\tilde{\mathbf{X}}^{\text{prop}})$ with the proposal density $q(\mathbf{x}^{\text{prop}}|\mathbf{x}) = |g'(\mathbf{x}^{\text{prop}})|\tilde{q}(\tilde{\mathbf{x}}^{\text{prop}}|\tilde{\mathbf{x}})$.