Tunings for leapfrog integration of Hamiltonian Monte Carlo for estimating genetic parameters

Aisaku Arakawa,¹ Takeshi Hayashi,² Masaaki Taniguchi,¹ Satoshi Mikawa,¹ Motohide Nishio¹

¹Division of Animal Breeding and Reproduction Research, Institute of Livestock and Grassland Science, National Agriculture and Food Research Organization (NARO), 2 Ikenodai, Tsukuba, Ibaraki, 305-0901, Japan;

²Division of Basic Research, Institute of Crop Science, NARO, 3-1-1 Kannondai, Tsukuba, Ibaraki, 305-8666, Japan.

Running head: Parameters estimated by HMC

Corresponding author: Aisaku Arakawa

Division of Animal Breeding and Reproduction Research, Institute of Livestock and Grassland Science, NARO, 2 Ikenodai, Tsukuba, Ibaraki, 305-0901, Japan

Tel: +81-29-838-8627

E-mail: aisaku@affrc.go.jp

Abstract

A Hamiltonian Monte Carlo algorithm is a Markov Chain Monte Carlo method that is considered more effective than the conventional Gibbs sampling method. Hamiltonian Monte Carlo is based on Hamiltonian dynamics, and it follows Hamilton's equations, which are expressed as two differential equations. In the sampling process of Hamiltonian Monte Carlo, a numerical integration method called leapfrog integration is used to approximately solve Hamilton's equations, and the integration is required to set the number of discrete time steps and the integration stepsize. These two parameters require some amount of tuning and calibration for effective sampling. In this study, we applied the Hamiltonian Monte Carlo method to animal breeding data and identified the optimal tunings of leapfrog integration for normal and inverse chi-square distributions. Then, using real pig data, we revealed the properties of the Hamiltonian Monte Carlo method with the optimal tuning by applying models including variance explained by pedigree information or genomic information. Compared with the Gibbs sampling method, the Hamiltonian Monte Carlo method had superior performance in both models. We have provided the source codes of this method written in the R language.

Keywords: Hamiltonian Monte Carlo, leapfrog integration, mixed model, genomic selection, Gibbs sampling

Background

Computing performance has rapidly improved in recent years, and Bayesian approaches have become more popular tools for estimating genetic parameters or predicting genomic breeding values in animal and plant breeding (Meuwissen et al. 2000; Jannink et al. 2010). In particular, Bayesian inferences have been used as an alternative method of a restricted maximum likelihood (REML) method to estimate parameters if analytical models are too complicated to apply REML (Sorensen et al. 1995; Meuwissen et al. 2000; Ibáñez-Escriche et al. 2008). In Bayesian inferences, the joint posterior distribution of all parameters is constructed by multiplying a likelihood that generates the data and prior distributions, and the marginal distributions of each parameter are obtained by integrating out all other parameters of interest from the joint posterior distribution. However, the most critical limitation of an application for Bayesian approaches in quantitative genetics is that a Bayesian calculation for marginal posterior distributions often requires integration for high-dimensional distributions, and it is difficult to estimate the parameters of interest using such an analytical calculation of complex integration.

Since a series of papers of Wang et al. (1993; 1994) was published in the field of animal breeding, GS has become increasingly a popular tool for estimating genetic parameters. The GS methods have several advantages compared with the REML method, and especially, if the size of data is too large or if the models are too complex for the REML method to handle, the GS method offers, in fact, a way to estimate genetic parameters, and it can provide an effective solution by generating successive samples from conditional posterior distributions. Alternatively, in the genomic era, the GS method has been used for most Bayesian alphabet algorithms (BayesA by Meuwissen et al. (2000), BayesC by Habier et al. (2011)) for estimating single nucleotide polymorphism (SNP) effects. In most cases, the GS methods employ a single-site sampling algorithm for estimating parameters because the

algorithm needs no inversion of the coefficient matrix of mixed model equations (Wang et al. 1994). Conversely, the GS method is implicitly known to require a long Markov Chain Monte Carlo (MCMC) chain to evaluate estimates of parameters of interest because the samples analyzed using the GS method are highly autocorrelated with each other, leading to a long computation time. Many researchers attempted to reduce the autocorrelations between samples using several matrix techniques under the GS scheme (García-Cortés and Sorensen 1996; Waldmann et al. 2008; Runcie and Mukherjee 2013).

Recently, the Hamiltonian Monte Carlo (HMC) algorithm has become a more popular tool in a Bayesian inference, which is based on Hamiltonian dynamics in physics (Neal 2011). The HMC algorithm was originally proposed by Duane et al. (1987) to apply the numerical simulation of lattice field. The HMC algorithm introduces an alternative variable or vector, which is called kinetic energy, to effectively transit samples within a parameter space; so, the HMC methods have a potential for giving a better sampling property than the GS methods. While the HMC algorithm could theoretically generate samples from a wide range of the parameter space with high probability, this sampling efficiency strongly depends on tunings for an approximation path integration method, a so-called leapfrog integration; the number of steps L and the stepsize ϵ . The stepsize ϵ governed the stability of the Hamiltonian function; for example, a larger stepsize than expected leads to a low acceptance ratio due to an increase of the integration error by the leapfrog integration. The number of steps L affects sampling efficiency; if L is not large enough, the samples generated by HMC show quite high autocorrelations between successive iterations, whereas if L is too large, the path approximated by leapfrog integration would retrace its previous steps of the initial state, which leads to wasted computing time (Neal 2011; Betancourt et al., 2017). Neal (2011) recommended that one practical solution for using HMC is to determine the length of trajectory, which requires selecting suitable values for L and ϵ in the leapfrog process.

However, in this case, we need a lot of preliminary runs with trial values for L and ϵ , and trace plots of the preliminary runs must be checked to determine how well these runs work.

In this study, we aimed to identify the suitable values for L and ϵ for leapfrog integration to optimize the HMC method for a linear mixed model. First, we derived the HMC algorithm by applying to estimate variance components to predict breeding values, and then we searched for optimal tunings for optimizing HMC. Finally, we demonstrated the computational properties of the HMC algorithm with the optimal values of L and ϵ using real pig data.

HMC

First, we will briefly introduce the HMC method. The HMC method is based on Hamiltonian dynamics, and the Hamiltonian (H) is expressed as

$$H(\theta, \mathbf{p}) = U(\theta) + K(\mathbf{p}), \quad (1)$$

where $U(\theta)$ and $K(\mathbf{p})$ are “potential” and “kinetic” energies, respectively, in a physical system. The kinetic energy term is expressed as $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}'\mathbf{M}^{-1}\mathbf{p}$, where $\mathbf{p}$ and $\mathbf{M}$ are interpreted as momentum variables and a mass matrix, respectively.

When estimating a random variable θ with density $p(\theta)$ using the HMC method, the independent auxiliary variable $\mathbf{p}$ is introduced. Its density is assumed to be normally distributed as follows: $p(\mathbf{p}) \sim N(0, \mathbf{M})$, where $\mathbf{M}$ is interpreted as a covariance matrix in statistics. The joint density $p(\theta, \mathbf{p})$ is expressed as $p(\theta)p(\mathbf{p})$ because of its independence. We denoted the logarithm form of the joint density as

$$\begin{aligned} p(\theta, \mathbf{p}) &= \exp[\log p(\theta) + \log p(\mathbf{p})] \\ &\propto \exp\left[\log p(\theta) - \frac{1}{2}\mathbf{p}'\mathbf{M}^{-1}\mathbf{p}\right]. \end{aligned} \quad (2)$$

The bracketed term in equation (2) is rewritten as

$$H(\theta, \mathbf{p}) = -\log p(\theta) + \frac{1}{2} \mathbf{p}' \mathbf{M}^{-1} \mathbf{p}, \quad (3)$$

which can be interpreted as H with potential energy

$$U(\theta) = -\log p(\theta),$$

and kinetic energy

$$K(\mathbf{p}) = \frac{1}{2} \mathbf{p}' \mathbf{M}^{-1} \mathbf{p}.$$

Hamilton's equations are known as partial derivatives of H with respect to fictitious time t ,

and according to Neal (2011), the equations are expressed as

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial \mathbf{p}} = \frac{\partial K}{\partial \mathbf{p}}, \quad (4)$$

and

$$\frac{d\mathbf{p}}{dt} = \frac{\partial H}{\partial \theta} = -\frac{\partial U}{\partial \theta}. \quad (5)$$

Hamiltonian dynamics has two important properties, namely, *reversibility* and *volume preservation* (Neal 2011), which rely on the use of MCMC updates. When an exact analytic solution of the differential equations (4) and (5) for Hamiltonian dynamics is available, we can use the proposed trajectory with the same volume of H . For practical applications, however, there is no analytic solution for Hamilton's equations, and therefore, Hamilton's equations must be approximated by discretizing time. The leapfrog discretization integration, also called the Stormer-Verlet method, provides a good approximation for Hamiltonian dynamics (Neal 2011). The leapfrog method depends on two arbitrarily inputted parameters, namely, L (the number of discrete time steps in leapfrog integration) and ϵ (the integration stepsize, indicating how far each leapfrog step jumps). Leapfrog integration is described as

$$\mathbf{p}\left(t + \frac{\epsilon}{2}\right) = \mathbf{p}(t) - \left(\frac{\epsilon}{2}\right) \frac{\partial U}{\partial \theta}(\theta(t)), \quad (6)$$

$$\theta(t + \epsilon) = \theta(t) + \epsilon \frac{\partial K}{\partial \mathbf{p}}\left(\mathbf{p}\left(t + \frac{\epsilon}{2}\right)\right), \quad (7)$$

$$\mathbf{p}(t + \epsilon) = \mathbf{p}\left(t + \frac{\epsilon}{2}\right) - \left(\frac{\epsilon}{2}\right) \frac{\partial U}{\partial \theta}(\theta(t + \epsilon)). \quad (8)$$

Hamiltonian dynamics is simulated using the leapfrog integration of equations (6–8) for L times ($0 < t < L$). Figure 1 shows the example for the Hamiltonian dynamics approximated by leapfrog integration. The horizontal axis is a potential variable which equals to a random variable of interest, and the vertical axis is a momentum variable sampled from a normal distribution. Each step between the consecutive t 's on Figure 1 corresponds to equations 6 to 8, and ϵ is expressed as a difference in distance between the consecutive t 's. The preservation of volume via Hamiltonian dynamics keeps H invariant, but H is not exactly conserved with the leapfrog method because of the integration error caused by the time discretization. Therefore, a Metropolis correction step is necessary to ensure correct sampling from the marginal distribution. In the Metropolis step, the new proposal samples $(\theta^*, \mathbf{p}^*)$ are accepted with probability

$$\alpha = \min \left[1, \frac{\exp(-H(\theta^*, \mathbf{p}^*))}{\exp(-H(\theta_i, \mathbf{p}_i))} \right], \quad (9)$$

which corresponds to the usual MH acceptance probability, or samples of $(\theta_i, \mathbf{p}_i)$ keep the current $(\theta_{i+1}, \mathbf{p}_{i+1}) = (\theta_i, \mathbf{p}_i)$. In the sampling method using Hamiltonian dynamics, θ and $\mathbf{p}$ are independent, and, therefore, the HMC method will give the θ values sampled from these marginal distributions. If the integration error in H remains small during the integration, then the HMC approach will achieve a high level of acceptance probability (almost 1.0).

Linear mixed model using the HMC method

We employed a univariate linear mixed model as follows:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{Z}\mathbf{a} + \mathbf{e}, \quad (10)$$

where $\mathbf{y}$ is an observation vector of order $n \times 1$, $\mathbf{b}$ and $\mathbf{a}$ are location parameters with different prior distributions of orders $p \times 1$ and $q \times 1$, respectively, $\mathbf{e}$ is the residual error of order $n \times 1$, and $\mathbf{X}$ and $\mathbf{Z}$ are designed matrices of orders $n \times p$ and $n \times q$, respectively. The likelihood for the model and the prior distributions for $\mathbf{b}$ and $\mathbf{a}$ can be specified as

$\mathbf{y}|\mathbf{b}, \mathbf{a}, \sigma_e^2 \sim N(\mathbf{Xb} + \mathbf{Za}, \mathbf{I}\sigma_e^2)$, $\mathbf{b}|\sigma_b^2 \sim N(\mathbf{0}, \mathbf{I}\sigma_b^2)$ and $\mathbf{a}|\sigma_a^2 \sim N(\mathbf{0}, \mathbf{A}\sigma_a^2)$, respectively, where σ_b^2 , σ_a^2 and σ_e^2 are the variances for $\mathbf{b}$, $\mathbf{a}$ and $\mathbf{e}$, respectively, $\mathbf{I}$ is an identity matrix, and $\mathbf{A}$ is a variance-covariance matrix relating to $\mathbf{a}$. In this study, σ_b^2 was set to be a constant value, and the prior distributions for σ_a^2 and σ_e^2 are expressed as $\sigma_a^2|v_a, S_a^2 \sim \chi_a^{-2}(v_a, S_a^2)$ and $\sigma_e^2|v_e, S_e^2 \sim \chi_e^{-2}(v_e, S_e^2)$, respectively, where v_a and v_e are the degrees of freedom for σ_a^2 and σ_e^2 , respectively, and S_a^2 and S_e^2 are scale parameters for σ_a^2 and σ_e^2 , respectively. The joint distribution of parameters of the linear mixed model for the Bayesian form is expressed as

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

and the logarithm of the joint distribution is

$$\begin{aligned} \log p(\mathbf{b}, \mathbf{a}, \sigma_a^2, \sigma_e^2 | \mathbf{y}) &\propto \log[p(\mathbf{y} | \mathbf{b}, \mathbf{a}, \sigma_a^2, \sigma_e^2) p(\mathbf{b}, \mathbf{a}, \sigma_a^2, \sigma_e^2)] \\ &= \log p(\mathbf{y} | \mathbf{b}, \mathbf{a}, \sigma_e^2) + \log p(\mathbf{b} | \sigma_b^2) + \log p(\mathbf{a} | \sigma_a^2) \\ &\quad + \log p(\sigma_a^2 | v_a, S_a^2) + \log p(\sigma_e^2 | v_e, S_e^2) \\ &\propto -\frac{(\mathbf{y} - \mathbf{Xb} - \mathbf{Za})'(\mathbf{y} - \mathbf{Xb} - \mathbf{Za}) + v_e S_e^2}{2\sigma_e^2} - \frac{\mathbf{b}'\mathbf{b}}{2\sigma_b^2} - \frac{\mathbf{a}'\mathbf{A}^{-1}\mathbf{a} + v_a S_a^2}{2\sigma_a^2} \\ &\quad - \frac{2+q+v_a}{2} \log(\sigma_a^2) - \frac{2+n+v_e}{2} \log(\sigma_e^2). \end{aligned}$$

As we denote $\log[p(\mathbf{y} | \mathbf{b}, \mathbf{a}, \sigma_a^2, \sigma_e^2) p(\mathbf{b}, \mathbf{a}, \sigma_a^2, \sigma_e^2)]$ as a function f , we changed to

$$f_b \propto \frac{(\mathbf{y} - \mathbf{Za})'\mathbf{Xb}}{2\sigma_e^2} - \frac{\mathbf{b}'\mathbf{X}'\mathbf{Xb}}{2\sigma_e^2} - \frac{\mathbf{b}'\mathbf{b}}{2\sigma_b^2} \quad (11)$$

$$f_a \propto \frac{(\mathbf{y} - \mathbf{Xb})'\mathbf{Za}}{2\sigma_e^2} - \frac{\mathbf{a}'\mathbf{Z}'\mathbf{Za}}{2\sigma_e^2} - \frac{\mathbf{a}'\mathbf{A}^{-1}\mathbf{a}}{2\sigma_a^2} \quad (12)$$

$$f_{\sigma_a^2} \propto -\frac{2+q+v_a}{2} \log(\sigma_a^2) - \frac{\mathbf{a}'\mathbf{A}^{-1}\mathbf{a} + v_a S_a^2}{2\sigma_a^2} \quad (13)$$

and

$$f_{\sigma_e^2} \propto -\frac{2+n+v_e}{2} \log(\sigma_e^2) - \frac{(\mathbf{y} - \mathbf{Xb} - \mathbf{Za})'(\mathbf{y} - \mathbf{Xb} - \mathbf{Za}) + v_e S_e^2}{2\sigma_e^2}. \quad (14)$$

Partial derivatives for f with respect to each parameter ($\mathbf{b}$ or b_i , $\mathbf{a}$ or a_i , σ_a^2 , and σ_e^2 , where b_i is the

i th element of $\mathbf{b}$, and a_i is the i th element of $\mathbf{a}$) are expressed as follows:

$$\frac{\partial f}{\partial \mathbf{b}} \propto \frac{\mathbf{X}'(\mathbf{y}-\mathbf{Za})-\mathbf{X}'\mathbf{Xb}}{\sigma_e^2} - \frac{\mathbf{b}}{\sigma_b^2}, \quad (15)$$

or

$$\frac{\partial f}{\partial b_i} \propto \frac{\mathbf{x}_i'(\mathbf{y}-\mathbf{X}_{-i}\mathbf{b}_{-i}-\mathbf{Za})-\mathbf{x}_i'\mathbf{x}_ib_i}{\sigma_e^2} - \frac{b_i}{\sigma_b^2}, \quad (16)$$

$$\frac{\partial f}{\partial \mathbf{a}} \propto \frac{\mathbf{Z}'(\mathbf{y}-\mathbf{Xb})-\mathbf{Z}'\mathbf{Za}}{\sigma_e^2} - \frac{\mathbf{A}^{-1}\mathbf{a}}{\sigma_a^2}, \quad (17)$$

or

$$\frac{\partial f}{\partial a_i} \propto \frac{\mathbf{z}_i'(\mathbf{y}-\mathbf{Xb}-\mathbf{Z}_{-i}\mathbf{a}_{-i})-\mathbf{z}_i'\mathbf{z}_ia_i}{\sigma_e^2} - \frac{\mathbf{A}_i^{-1}\mathbf{a}}{\sigma_a^2}, \quad (18)$$

$$\frac{\partial f}{\partial \sigma_a^2} \propto -\frac{2+q+v_a}{\sigma_a^2} + \frac{\mathbf{a}'\mathbf{A}^{-1}\mathbf{a}+v_aS_a^2}{2(\sigma_a^2)^2}, \quad (19)$$

and

$$\frac{\partial f}{\partial \sigma_e^2} \propto -\frac{2+n+v_e}{\sigma_e^2} + \frac{(\mathbf{y}-\mathbf{Xb}-\mathbf{Za})'(\mathbf{y}-\mathbf{Xb}-\mathbf{Za})+v_eS_e^2}{2(\sigma_e^2)^2}, \quad (20)$$

where $\mathbf{b}_{-i}$ is the vector of $\mathbf{b}$ without b_i , $\mathbf{a}_{-i}$ is the vector of $\mathbf{a}$ without a_i , $\mathbf{x}_i$ is the i th column vector relating to b_i , $\mathbf{X}_{-i}$ is the matrix relating to $\mathbf{b}_{-i}$, $\mathbf{z}_i$ is the i th column vector relating to a_i , and $\mathbf{Z}_{-i}$ is the matrix relating to $\mathbf{a}_{-i}$. $\mathbf{A}_i^{-1}$ is the i th row vector of $\mathbf{A}^{-1}$. The HMC method could successively generate random samples from the joint posterior distribution by substituting equations (15 or 16, 17, or 18, 19, and 20) into $\frac{\partial U}{\partial \theta}(\theta(t))$ in equation (6) and $\frac{\partial U}{\partial \theta}(\theta(t+1))$ in equation (8). The pseudo-code for the HMC method with $\mathbf{M} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix, is shown in algorithm 1, and the R codes for the linear mixed model are written in Appendix III.

198 **Algorithm 1.** Hamiltonian Monte Carlo algorithm

1:	Input: Starting position $\theta_{current}$, stepsize ϵ , and discrete time steps L
2:	$\theta_0^{(t)} := \theta_{current}$
3:	$p_0^{(t)} \sim N(0,1)$ #Sample momentum variable from a normal distribution
4:	Calculate $H_0^{(t)}(\theta_0^{(t)}, p_0^{(t)})$ #Calculate Hamiltonian before leapfrog integration
5:	for $i = 1$ to L #Leapfrog integration
6:	$p_{i-\frac{1}{2}}^{(t)} \leftarrow p_{i-1}^{(t)} - \frac{\epsilon}{2} \frac{\partial U}{\partial \theta}(\theta_{i-1}^{(t)})$ #substituting each of the derivative equations ([15], [16], [17], [18], [19], or [20]) into $\frac{\partial U}{\partial \theta}(\theta_{i-1}^{(t)})$ in equation [6]
7:	$\theta_i^{(t)} \leftarrow \theta_{i-1}^{(t)} + \epsilon p_{i-\frac{1}{2}}^{(t)}$ #equation [7]
8:	$p_i^{(t)} \leftarrow p_{i-\frac{1}{2}}^{(t)} - \frac{\epsilon}{2} \frac{\partial U}{\partial \theta}(\theta_i^{(t)})$ #substituting each of the derivative equations ([15], [16], [17], [18], [19], or [20]) into $\frac{\partial U}{\partial \theta}(\theta_i^{(t)})$ in equation [8]
9:	End
10:	Calculate $H_L^{(t)}(\theta_L^{(t)}, p_L^{(t)})$ #Calculate Hamiltonian after leapfrog integration
11:	$u \sim Uniform[0,1]$ #MH correlation
13:	if $u < \min\left(1, \exp\left[H_L^{(t)}(\theta_L^{(t)}, p_L^{(t)}) - H_0^{(t)}(\theta_0^{(t)}, p_0^{(t)})\right]\right)$, then
14:	$\theta_{current} = \theta_L^{(t)}$
15:	end if

199

Properties of and optimization for leapfrog integration

An HMC algorithm strongly depends on choosing two parameters of leapfrog integration; L and ϵ . In a simple situation of leapfrog integration, a trajectory of one sample shows periodically a trace on an elliptical trajectory in two dimensions (Neal 2011), as in Figure 1. We measured several steps for one round of the trajectory using leapfrog integration (L_{one_round} in Figure 1) to investigate the properties of the leapfrog integrations for normal and inverse chi-square distributions. A total length of the trajectory in two dimensions is expressed as ϵL_{one_round} (Figure 1).

Let x and v be a sample from a normal distribution $N(\mu, \sigma^2)$ and a scaled inverse chi-square distribution $\chi^{-1}(n, \mathbf{u}'\mathbf{u})$, respectively, where n is a value of a degree of belief. Logarithm forms of these distributions are described as follows:

$$f(x|\mu, \sigma^2) \propto -\frac{(x-\mu)^2}{2\sigma^2} \quad (21)$$

and

$$f(v|n, \mathbf{u}'\mathbf{u}) \propto -\frac{2+n}{2} \log(v) - \frac{\mathbf{u}'\mathbf{u}}{2v}, \quad (22)$$

where $\mathbf{u}'\mathbf{u}$ in equation (22) is expressed as $(n-1)E[v]$ and $E[v]$ is an expectation value of v .

Variances of the normal and inverse chi-square distributions are σ^2 and $\frac{2(\mathbf{u}'\mathbf{u})^2}{(n-2)^2(n-4)}$,

respectively. The partial derivatives for the normal and inverse chi-square distributions with respect to the parameters x and v are

$$\frac{\partial f(x|\mu, \sigma^2)}{\partial x} = -\frac{x-\mu}{\sigma^2} \quad (23)$$

and

$$\frac{\partial f(v|n, \mathbf{u}'\mathbf{u})}{\partial v} = -\frac{2+n}{2v} + \frac{\mathbf{u}'\mathbf{u}}{2v^2}, \quad (24)$$

respectively. According to equations (6–8), the leapfrog integration steps might be influenced by μ and σ^2 for the normal distribution and n and $\mathbf{u}'\mathbf{u}$ for the inverse chi-square distribution.

The tests were run using setting at different values of μ (0 and 100) and σ^2 (1, 10, and 100) for the normal distribution and different values of n (101, 1,001 and 10,001) and $\mathbf{u}'\mathbf{u}$ (1,000 and 100,000) for the inverse chi-square distribution.

Rao (1945) showed the relationship between Riemann geometry and statistics, and recently, Girolami et al. (2011) incorporated a Riemann manifold in HMC, which can explain the curvature of the conditional posterior distributions by Riemann geometry. Holmes et al. (2013) and Betancourt (2017) showed geometrical interpretations for HMC. According to Riemann geometry, the Riemannian metric is defined by the Fisher information (Amari 2016). According to the similar manner of the Fisher information, second-order derivatives of the normal and the inverse chi-square distributions are

$$\frac{\partial^2 f(x|\mu, \sigma^2)}{\partial x \partial x} = -\frac{1}{\sigma^2}, \quad (25)$$

and

$$\frac{\partial^2 f(v|n, \mathbf{u}'\mathbf{u})}{\partial v \partial v} = \frac{2+n}{2v^2} - \frac{\mathbf{u}'\mathbf{u}}{v^3}, \quad (26)$$

respectively. Substituting $v = \frac{\mathbf{u}'\mathbf{u}}{n-1}$ into the above equation (26), we obtained

$$\frac{\partial^2 f(v|n, \mathbf{u}'\mathbf{u})}{\partial v \partial v} = -\frac{(n-2)^2(n-4)}{2(\mathbf{u}'\mathbf{u})^2}. \quad (27)$$

The above equations (25) and (27) correspond to negative forms of inversed variances for each distribution. In this study, we chose a square root of these variances for the two distributions as a basic indicator of ϵ in order to clarify the influence of the size of ϵ on estimation by the HMC method. ϵ was set to be $\frac{\sqrt{var}}{\alpha}$, where $\sqrt{var}$ is the variance of either distribution, and α was set to 1, 10, and 100.

After deciding the optimal values of ϵ , we attempted to detect the influence of the number of L on the precision of estimates via the HMC method using the same simulation data by QMSim (Sargolzaei and Schenkel 2009). In the simulation, the degree of heritability

was assumed to be 0.50, and phenotypic variance was set at 1.0. The base population consisted of 5 males and 50 females, and these base animals were mated at random; specifically, one male in the base population was randomly mated with 10 females to produce two males and two females of generation 1. Five males were randomly selected as sires for the next generation and mated with 10 females to produce the next generation. Five discrete generations were simulated without the base population, and the data for the base population were removed. The population size was 1,000 with equal numbers of males and females. In total, 10,000 samples were simulated, of which the first 1,000 were discarded as burn-in iterations. The post-analysis for the sampling sequences was conducted using the “*effectiveSize*” function of the R “*coda*” package (Plummer et al. 2006) to estimate the effective sample sizes (ESSs) of the sequences. We compared estimating properties for variance components and breeding values using the GS method. The starting values were set at 0.5 for the variance components and 0 for the fixed and random effects in all of the analyses. The HMC and GS programs were written in Fortran 90. In the analysis of genetic variance explained by pedigree information, we employed a sparse matrix routine by Misztal (2014).

Application study for data

We applied the HMC method to a public pig dataset (Cleveland et al. 2012) using an infinitesimal animal model and investigated the properties of HMC sampling. In addition, we compared the HMC algorithm with GS, which is the conventional approach in Bayesian inference. According to Cleveland et al. (2012), we selected two traits that were expressed as t1 and t5 because the two traits have different genetic backgrounds; Cleveland et al. (2012) reported using the full dataset that the traits t1 and t5 are low ($h^2 = 0.07$) and high ($h^2 = 0.62$) heritabilities, respectively, and the phenotypic variance for t1 ($\sigma_p^2 = 3.14$) is quite lower

than that for t5 ($\sigma_p^2 = 5579.12$). We used phenotypic, pedigree, and genomic data. The numbers of recorded animals in t1 and t5 were 2,804 and 3,184, respectively, and pedigree information was stored for 6,473 animals. We used only SNP genotypes with a minor allele frequency of >0.05 ; the total number of SNPs was 45,385. We applied three single-trait models to estimate variance components; $\mathbf{y} = \mathbf{1}\mu + \mathbf{Z}\mathbf{a} + \mathbf{e}$, $\mathbf{y} = \mathbf{1}\mu + \mathbf{Z}\mathbf{g} + \mathbf{e}$, and $\mathbf{y} = \mathbf{1}\mu + \mathbf{Z}\mathbf{g} + \mathbf{Z}\mathbf{d} + \mathbf{e}$, where $\mathbf{a}$ is a vector of additive genetic effects, which is distributed $N(\mathbf{0}, \mathbf{A}\sigma_a^2)$; where $\mathbf{A}$ is an additive relationship matrix from pedigree information, where $\mathbf{g}$ is a vector of additive genomic effects that is distributed $N(\mathbf{0}, \mathbf{G}\sigma_g^2)$, where $\mathbf{G}$ is an additive genomic relationship matrix that is calculated as in the study by VanRaden (2008), $\mathbf{d}$ is a vector of dominance deviations that is distributed $N(\mathbf{0}, \mathbf{D}\sigma_d^2)$, and where $\mathbf{D}$ is a covariance matrix relating to $\mathbf{d}$ that is constructed based on the work of Vitezica et al. (2013). We used only t5 for applying the model, including the dominance variance, because the previous study reported by Da et al. (2014) showed that dominance variance for t1 was quite low. Overall, 110,000 samples were simulated, the first 10,000 of which were discarded as burn-in iterations. After the samples were generated, 10,000, 50,000, and 100,000 samples after the burn-in period were used to investigate the performance of the HMC method in comparison with that of the GS method. In a post-analysis for the sampling sequences using the two methods, the “*effectiveSize*” function of the R “*coda*” package (Plummer et al. 2006) was used to estimate ESSs of the sequences.

Results

Leapfrog integration

Tables 1 and 2 show a summary of the number of steps for one round of the trajectory via leapfrog integration (L_{one_round}) for the normal and the inverse chi-square distributions, respectively. For the normal distribution, when ϵ was expressed as the function of σ^2 , the

scales of μ had no influence on the number of steps per round of the trajectory, whereas the results in Table 1 showed a linear relationship between the size of σ^2 and L_{one_round} . Consequently, the total length of the trajectory by leapfrog integration (ϵL_{one_round}) was expressed as $\frac{\sqrt{\sigma^2}}{0.159}$. In practical application, we need the variance for conditional posterior distributions for fixed and random effects; so, Appendix I shows the deviations of σ^2 for fixed and random effects.

In the case of inverse chi-square distribution (Table 2), when the values of ϵ were assumed to be the function of its variance, the size of $\mathbf{u}'\mathbf{u}$ did not affect L_{one_round} because the variance is a function of $\mathbf{u}'\mathbf{u}$ ($\frac{2(\mathbf{u}'\mathbf{u})^2}{(n-2)^2(n-4)}$), whereas the values of n had a little influence on L_{one_round} , which means that if we set a higher degree for n , we need a longer distance of trajectory (847 in $n = 101$, and 889 in $n = 10001$ under $\alpha = 100$). However, the effect of the values of n was quite small; so, we obtained the total length of the trajectory by leapfrog integration (ϵL_{one_round}), which was expressed as $\frac{\sqrt{var}}{0.112}$, where var is $\frac{2(\mathbf{u}'\mathbf{u})^2}{(n-2)^2(n-4)}$.

Table 1. Number of steps for one round of leapfrog integration (L_{one_round}) under the normal distribution $N(\mu, \sigma^2)$ ($\epsilon = \sqrt{\sigma^2}/\alpha$)

α	$\mu = 0$			$\mu = 100$		
	$\sigma^2 = 1$	$\sigma^2 = 10$	$\sigma^2 = 100$	$\sigma^2 = 1$	$\sigma^2 = 10$	$\sigma^2 = 100$
1	6	6	6	6	6	6
10	63	63	63	63	63	63
100	629	629	629	629	629	629

Table 2. Number of steps for one round of leapfrog integration (L_{one_round}) under the inverse chi-square distribution $\chi^{-1}(n, \mathbf{u}'\mathbf{u})$ ($\epsilon = \sqrt{var}/\alpha$, where $var = 2(\mathbf{u}'\mathbf{u})^2 / ((n-2)^2(n-4))$)

α	$\mathbf{u}'\mathbf{u} = 1,000$			$\mathbf{u}'\mathbf{u} = 100,000$		
	$n = 101$	$n = 1001$	$n = 10001$	$n = 101$	$n = 1001$	$n = 10001$
1	9	9	9	9	9	9
10	85	88	88	85	88	88
100	847	885	889	847	885	889

Inference of the discretizing time

We chose 20 on L_{one_round} in order to determine the optimal value for L , and ϵ s for the normal and inverse chi-square distributions were expressed as $\frac{\sqrt{\sigma^2}}{0.159 \times 20}$ and $\frac{\sqrt{var}}{0.112 \times 20}$. L was changed from 1 to 20, and the results were conducted by averaging the five different MCMC chains with a different seed. The results of the summary for variance components and breeding values for each L using the HMC and GS methods are shown in Table 3. The acceptance ratios were almost one in all cases, and posterior estimates using the two methods were identical to each other for all values of L excluding 10 and 20. The ESSs using the HMC with for L values of 5, 6, 8, 9, 11, 12, 13, 14, and 15 were higher than those for GS. However, in the case of $L = 10$, the samples of the breeding values using the HMC method had extremely high ESS values ($67,267.7 \pm 237,508.4$). A sampling sequence of the breeding value of the animal with the largest ESS is presented in Figure 2. The graphic illustrates that the samples for the breeding value had a cyclical periodicity along the sampling sequence (Figure 2a), and that the changes of autocorrelations between lags were drastic between plus and minus (Figure 2b), suggesting that breeding values could not be sampled randomly in the case of $L = 10$. This nonrandom sampling for the breeding values leads to quite high estimates of genetic variance (1.39 ± 0.11).

[Insert Table 3]

Real data

We applied the HMC method with $L = 7$ on $L_{one_round} = 20$ to the real pig data, and we also performed the analysis using 10,000, 50,000, and 100,000 samples. Summary statistics for the marginal distributions of variance components for t1 and t5 using pedigree information are shown in Tables 4 and 5, respectively, and the marginal posterior

distributions for t1 and t5 are shown in Figures 3 and 4, respectively. The posterior statistics obtained using the HMC method were similar to those obtained using GS for the two traits, in line with results reported by Cleveland et al. (2012) using full pedigree information. The ESS values for the two methods generally increased linearly with an increase in the sample size. For both traits, most of the ESSs for the two variances using the HMC method were more than those using the GS method. We compared the marginal posterior distributions using the HMC method for t1 (Figure 3) and t5 (Figure 4). Excluding the genetic variances for t1, all of the variances depicted using 50,000 or 100,000 samples were similar to each other (c vs. d in Figure 3; a vs. b and c vs. d in Figure 4). In the case of 10,000 samples, the variances depicted using the HMC method were similar to those for larger sample numbers, whereas the GS method demonstrated figures that were slightly different from those for larger sample numbers. For genetic variances for t1 (Figure 3a and 3b), the marginal posterior distributions obtained using 10,000 samples exhibited polymodality and a lack of smoothness for both methods, and for the GS method, the marginal posterior distributions were also bimodal and less smooth even if 100,000 samples were generated. On the contrary, the HMC method produced a unimodal distribution that was smoother than that produced using the GS method.

Table 4. Summary statistics of variance components for trait 1 (t1) using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods with 10,000, 50,000, and 100,000 sampling sequences

Method	Sample Length	Residual Variance					Genetic Variance				
		Mean	Median	Mode	SD	ESS	Mean	Median	Mode	SD	ESS
HMC	10,000	1.35	1.35	1.35	0.05	40.2	0.11	0.11	0.10	0.05	10.8
	50,000	1.35	1.35	1.34	0.05	171.1	0.11	0.11	0.09	0.04	55.4
	100,000	1.35	1.35	1.35	0.05	257.0	0.11	0.11	0.09	0.04	101.3
GS	10,000	1.36	1.36	1.35	0.03	69.3	0.11	0.11	0.11	0.03	17.2
	50,000	1.37	1.37	1.37	0.05	60.9	0.09	0.09	0.10	0.04	23.9
	100,000	1.36	1.36	1.30	0.05	142.4	0.10	0.10	0.07	0.04	57.6

ESS, effective sample size.

Table 5. Summary statistics of the variance components for trait 5 (t5) using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods with 10,000, 50,000, and 100,000 sampling sequences

Method	Sample Length	Residual Variance					Genetic Variance				
		Mean	Median	Mode	SD	ESS	Mean	Median	Mode	SD	ESS
HMC	10,000	1957.6	1956.7	1962.6	105.9	366.2	1579.3	1577.9	1581.9	148.6	192.8
	50,000	1958.5	1957.5	1963.3	106.7	1610.7	1576.7	1575.2	1577.1	148.0	905.9
	100,000	1959.1	1958.0	1959.1	106.0	3391.1	1574.4	1572.0	1573.0	145.7	1871.1
GS	10,000	1954.8	1953.4	1961.4	95.5	204.8	1579.6	1575.7	1571.3	138.2	122.7
	50,000	1953.5	1953.1	1953.1	92.1	976.6	1582.2	1579.0	1579.1	130.4	686.6
	100,000	1954.0	1953.7	1954.0	91.5	1962.4	1581.5	1578.0	1580.1	128.4	1371.6

ESS, effective sample size.

We compared two methods applied to the same pig data but replaced the genomic information. We also performed the analysis generating 10,000, 50,000, and 100,000 samples. Summary statistics for the marginal distributions of variance components for t1 and t5 are shown in Tables 6 and 7, respectively, and the marginal posterior distributions for t1 and t5 are shown in Figures 5 and 6, respectively. The posterior statistics obtained by using the HMC method were similar to those obtained using GS for the two traits. Also, the ESS values for the two methods increased linearly with an increase in the sample size, and the ESSs for the two variances using the HMC method were much higher than those obtained by using the GS method. Compared with the marginal posterior distributions, in the trait t5, the marginal posterior distributions of all variances in the two methods were quite similar despite the sampling size (Figure 6). However, in the trait t1, the marginal distributions for genetic variances depicted using the GS method were extremely skewed even if 100,000 samples were used (b in Figure 5).

Table 6. Summary statistics of variance components for trait 1 (t1) using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods with 10,000, 50,000, and 100,000 sampling sequences under genomic information.

Method	Sample Length	Residual Variance					Genomic Variance				
		Mean	Median	Mode	SD	ESS	Mean	Median	Mode	SD	ESS
HMC	10,000	1.42	1.43	1.42	0.04	134.9	0.03	0.02	0.02	0.02	5.1
	50,000	1.42	1.42	1.42	0.04	648.1	0.04	0.03	0.02	0.02	36.7
	100,000	1.42	1.42	1.42	0.04	1342.2	0.04	0.04	0.03	0.02	91.1
GS	10,000	1.42	1.42	1.42	0.03	177.0	0.04	0.04	0.03	0.02	7.7
	50,000	1.44	1.44	1.44	0.03	263.2	0.02	0.01	0.01	0.02	11.5
	100,000	1.43	1.43	1.44	0.03	350.5	0.02	0.01	0.01	0.02	24.8

ESS, effective sample size.

Table 7. Summary statistics of the variance components for trait 5 (t5) using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods with 10,000, 50,000, and 100,000 sampling sequences under genomic information.

Method	Sample Length	Residual Variance					Genomic Variance				
		Mean	Median	Mode	SD	ESS	Mean	Median	Mode	SD	ESS
HMC	10,000	2161.2	2160.0	2155.9	76.1	1165.6	1322.7	1315.6	1301.5	126.9	267.2
	50,000	2161.4	2160.4	2153.3	75.6	6366.6	1318.9	1314.5	1299.5	122.1	1600.4
	100,000	2161.7	2160.7	2154.9	75.4	12142.9	1318.1	1314.0	1293.7	123.3	3086.1
GS	10,000	2160.7	2160.1	2155.5	64.4	435.7	1316.5	1309.4	1300.6	119.3	183.0
	50,000	2159.4	2158.8	2153.6	63.5	2388.0	1321.0	1315.3	1294.8	113.6	970.3
	100,000	2161.3	2160.9	2156.9	63.4	4519.7	1312.1	1312.1	1293.6	113.3	1973.0

ESS, effective sample size.

387 In more complex situations, such as in the models including non-additive genetic
388 effects like dominance deviations, the summary statistics for the marginal distributions of
389 variance components for t5 are shown in Table 8, and the marginal posterior distributions are
390 shown in Figure 7. In the dominance variance, using the HMC method produced similar
391 estimates despite the sample size; however, the estimates of the GS method appeared unstable
392 while generating 100,000 samples and were lower than those obtained using the HMC method.
393 The marginal distributions of the dominance variance using the GS method were slightly
394 skewed compared with those obtained using the HMC method.

395 **Table 8.** Summary statistics of variance components for trait 1 (t5) using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling methods
396 with 10,000, 50,000, and 100,000 sampling sequences

Method	Sample Length	Residual Variance					Additive Genomic Variance					Dominance Genomic variance				
		Mean	Median	Mode	SD	ESS	Mean	Median	Mode	SD	ESS	Mean	Median	Mode	SD	ESS
HMC	10,000	1981.3	1981.4	1975.7	87.3	154.3	1313.5	1309.3	1293.2	125.7	281.5	186.4	178.9	166.6	57.7	28.0
	50,000	1982.6	1981.0	1975.7	91.4	602.6	1308.3	1304.2	1291.5	122.7	1469.1	186.1	182.8	167.2	65.0	93.1
	100,000	1986.6	1985.0	1980.9	92.3	1033.2	1306.4	1302.2	1289.0	123.7	3077.6	182.8	180.6	172.7	66.2	201.3
Gibbs sampling	10,000	2007.4	2008.2	2008.4	80.9	78.9	1302.3	1299.5	1298.3	111.4	193.2	156.7	155.2	160.6	63.4	13.1
	50,000	2022.6	2021.6	2021.8	82.5	324.9	1296.6	1294.6	1289.6	111.0	972.6	144.0	143.7	161.1	63.1	59.0
	100,000	2010.0	2008.8	2002.5	83.2	688.5	1300.9	1297.8	1293.6	114.1	1827.3	156.3	157.4	166.5	63.1	129.2

397 ESS, effective sample size.

We compared the computation times of both methods, using MacBook pro on an Intel Core i7 Processor (2.7 GHz) with 16 GB of RAM. We generated five chains with different seeds for the trait t1 using the model, including the genomic additive variance. In HMC, each simulation took $5,957.8 \pm 12.6$ sec, while the corresponding average time for GS was $5,924.1 \pm 4.1$ sec. In the HMC iteration, the leapfrog integrations of fixed and random effects are decomposed into the right and left hands of Henderson's mixed model equations (Appendix II), which include several matrix-vector multiplications. The same multiplications are needed in the GS iteration. These matrix-vector multiplication calculations are heavier than the leapfrog integration in HMC or a random number generator in GS. Consequently, the total computational time is quite similar to each other.

Discussion

Bayesian statistics have provided large amounts of information, and the GS method is the conventional Bayesian approach in animal breeding, being a feasible procedure for constructing the posterior distribution of interest. In this study, we proposed another MCMC method, called HMC, which is based on Hamilton dynamics, for estimating genetic parameters in the animal breeding fields. The HMC method also requires consideration of sampling convergence, length of the burn-in period, the number of samples, and ESS similar to other MCMC methods. Recently, many complex models, such as random regression models (Jamrozik and Schaeffer 1997) as well as a single-step genomic best linear unbiased prediction (Aguilar et al. 2010), have been proposed for animal breeding analyses. The likelihood functions of these analyses are likely too complex to be performed using REML. Although the GS method can provide marginal posterior distributions, in the case of analyzing complex models for which REML is inapplicable, GS exhibits extremely slow convergence and generates highly autocorrelated samples. Contrarily, the HMC method uses gradient

information about a logarithm of a posterior distribution to investigate the distribution space, which may lead to better mixing properties than the MH and GS methods.

The HMC method is an efficient sampling method, but the sampling performance of the HMC method strongly depended on the leapfrog integration parameters L and ϵ (Neal et al. 2011). The leapfrog integration process given a relatively large ϵ could not approximate the path for the trajectory adequately during the discretization time, and if using a small ϵ , more time would be needed to approximate the distance of a trajectory via leapfrog integration. If we set the same L in the HMC method, we could not obtain samples from marginal distribution under the large ϵ , while the samples are similar to those of the previous iteration under the small ϵ . In our study, we tried to reveal the properties of leapfrog integration for the normal and inverse chi-square distributions. When one parameter is considered, a trajectory of an approximate path using leapfrog integration according to equations (6–8) shows an ellipse on two dimensions. In order to optimize the performance of the HMC method, we decided on the length of the trajectory of the normal and the inverse chi-square distributions as the value of ϵL_{one_round} . When L_{one_round} was constant at 20 as a maximum discretization time, an L value of 7 provided a good performance for the HMC method. However, our settings for the HMC method were only used in the model, including random effects with no correlations. Therefore, we would need a modification to handle the correlation parameters, such as genetic correlations.

The parameter space of a statistical model can be expressed as a Riemann manifold, which can define the structure of the posterior distribution geometrically (Rao 1945). Girolami and Calderhead (2011) showed a more excellent way of incorporating the MHC algorithm into Riemann geometry in order to address many of the shortcomings of HMC. This algorithm is called Riemannian Manifold HMC (RMHMC), which can explain the curvature of the conditional posterior distributions by Riemann geometry. In this theory, an

information matrix $\mathbf{G}(\theta)$ is used instead of a fixed mass matrix $\mathbf{M}$ of the kinetic energy term $K(\mathbf{p})$, and the kinetic energy term is modified as

$$K(\mathbf{p}) = \frac{1}{2} \mathbf{p}' \mathbf{G}(\theta)^{-1} \mathbf{p}.$$

Girolami and Calderhead (2011) used the expected Fisher information matrix as $\mathbf{G}(\theta)$, which is defined as a positive semidefinite, whereas Paquet and Fraccaro (2016) used the observed Fisher information matrix. Although our results partially related to the Riemannian manifold, compared with these studies, we assigned the square root of variances of the conditional distributions to ϵ rather than $\mathbf{M}$ of the kinetic energy term. In addition, the variances of the conditional distributions do not correspond completely to the Fisher information. Our approach projected Hamiltonian function onto a Euclidean manifold and did not fully consider the local structure of the target distribution. Therefore, our approach would not be able to fully guarantee sampling from the marginal distributions within a parameter space when true values are on the edge of the parameter space. We applied the HMC with our tunings to extreme simulated data (10 individuals, $h^2 = 0$, and 10,000 SNP markers with equilibrium). We generated 10 different datasets with different seeds. We analyzed using the model including an additive and dominance genomic variance. As a result, the HMC method with our tunings did not fail outside of the parameter spaces for the variance components and breeding values, suggesting that our turnings could estimate parameters on the edge of the parameter space without failure (data not shown).

Our approach has two advantages compared with the RMHMC algorithm. First, it was easy to apply the HMC algorithm to a single trait linear mixed model because we only use

$$\frac{\sqrt{\sigma^2}}{0.159 \times L_{one_round}} \text{ and } \frac{\sqrt{var}}{0.112 \times L_{one_round}}$$

respectively, as ϵ . Second, in the RMHMC algorithm, the Fisher information and a first-order derivative of the Fisher information are needed in the leapfrog process. Therefore, potential

energy is no longer independent from kinetic energy in RMHMC; hence, fixed point iterations must be employed in the process of leapfrog integration of RMHMC, which suggests that RMHMC needs more nested iterations within leapfrog integration.

In comparison with the computing times of the GS methods, theoretically, the HMC method requires a longer computing time for the discrete time steps (L) than the GS method, but the HMC method showed similar computing time with the GS method in the context of genomic analysis. As previously mentioned, in the context of the mixed model, both HMC and GS require the same times of matrix-vector multiplications in the sampling of fixed and random effects in each iteration, which involves heavy computation within MCMC iterations.

The HMC method gave a higher ESS compared with the GS method, and the samples from the HMC methods could be generated from a wider range of its sampling space. Therefore, it would be possible for the HMC methods to shorten the total sample size, which leads to markedly decreasing the total computing time in HMC. Furthermore, in this study, the HMC method showed better sampling properties in the case of low heritability than those of the GS method because the HMC method gave a relatively smooth marginal distribution even in low heritability (the additive genetic variances in Figures 3 and 5 and the dominance genetic variances in Figure 7).

Many HMC algorithms have been developed to be free of problems concerning leapfrog integration and to shorten the burn-in period or accelerate its mixing properties. The most popular algorithm is No-U-Turn-Sampler (NUTS) (Hoffman and Gelman 2014), and STAN software (Carpenter et al. 2017), which has rapidly gained popularity in many Bayesian analysis fields, is equipped with this algorithm. The NUTS algorithm is extremely effective for the sampling process because it automates tuning in leapfrog integration, as neither the step size nor the number of steps needs to be specified by the user. However, the algorithm has a severe disadvantage regarding computing time because NUTS needs to

construct a deep binary tree in each step to specify an optimal L value. Additionally, STAN is a stand-alone program, and thus, it is difficult to modify it to analyze large amounts of animal breeding data as those generated for estimating genomic evaluation. Compared with our optimized HMC method, we need to set the leapfrog tuning beforehand, and the numbers of steps and stepsizes on our algorithm are not determined successively for each transition in each iteration.

In this study, we developed the HMC algorithm for a simple mixed model and optimized the algorithm to enable effective sampling from marginal posterior distributions. HMC could be generalized to more complex situations, such as a multiple-traits model (Van Tassel and Van Vleck 1996) or a threshold model (Sorensen et al. 1995), but we need to identify another optimization parameter of leapfrog integration for covariance components or thresholds; contrarily, a more flexible algorithm, such as RMHMC, must be applied to these generalized models.

Conclusion

In this study, we examined the HMC algorithm in the context of a linear mixed mode on quantitative genetics. This method strongly depends on two parameters, ϵ and L , for leapfrog integration. We applied one of the tunings for the integration process. The HMC method with optimized tuning provided superior sampling performances compared to the GS method. In addition, the HMC method appeared to generate samples from a wider range of parameter spaces than the GS method. The complete R and Fortran scripts are available from Aisaku Arakawa on reasonable request.

Acknowledgments

The authors thank Dr. Andres Legarra at INRA Toulouse for his constructive comments on an earlier manuscript version. A.A. conducted a portion of this work while visiting INRA Toulouse.

Funding

This study was supported by the research grant of the National Agricultural Research Organization (NARO).

Appendixes

Appendix I

According to Wang et al. [10], the factorization form of i th fixed effect (f_{b_i}) and i th breeding values (f_{a_i}) can be expressed as

$$f_{b_i} = -\frac{(b_i - \mu_b)' V_b^{-1} (b_i - \mu_b)}{2} - C_{-b}, \quad (\text{A1})$$

and

$$f_{a_i} = -\frac{(a_i - \mu_a)' V_a^{-1} (a_i - \mu_a)}{2} - C_{-a}, \quad (\text{A2})$$

where b_i and a_i are the i th fixed effect and i th breeding value, respectively, C_{-b} and C_{-a} are components not including elements related to b_i and a_i , respectively, μ_b and μ_a are described as

$$\mu_b = \left(\frac{\mathbf{x}_i' \mathbf{x}_i}{\sigma_e^2} + \frac{1}{\sigma_b^2} \right)^{-1} \left(\frac{\mathbf{x}_i' (\mathbf{y} - \mathbf{X}_{-i} \mathbf{b}_{-i} - \mathbf{Z} \mathbf{a})}{\sigma_e^2} \right),$$

and

$$\mu_a = \left(\frac{\mathbf{z}_i' \mathbf{z}_i}{\sigma_e^2} + \frac{\mathbf{A}_{ii}^{-1}}{\sigma_a^2} \right)^{-1} \left(\frac{\mathbf{z}_i' (\mathbf{y} - \mathbf{X} \mathbf{b} - \mathbf{Z}_{-i} \mathbf{a}_{-i})}{\sigma_e^2} - \sum_{k=1, k \neq i}^q \frac{\mathbf{A}_{ik}^{-1}}{\sigma_a^2} \right),$$

respectively, and V_b and V_a are described as

$$544 \quad V_b = \left(\frac{\mathbf{x}_i' \mathbf{x}_i}{\sigma_e^2} + \frac{1}{\sigma_b^2} \right)^{-1} \quad (A3)$$

545 and

$$546 \quad V_a = \left(\frac{\mathbf{z}_i' \mathbf{z}_i}{\sigma_e^2} + \frac{\mathbf{A}_{ii}^{-1}}{\sigma_a^2} \right)^{-1}, \quad (A4)$$

547 respectively. The two factorization forms (A1) and (A2) can be regarded as normal

548 distributions of $b_i | \mu_b, V_b \sim N(\mu_b, V_b)$ and $a_i | \mu_a, V_a \sim N(\mu_a, V_a)$.

Appendix II

We can change equations (15–18) of leapfrog integration, and the equations are expressed as follows:

$$\frac{\partial f}{\partial \mathbf{b}} \propto \frac{1}{\sigma_e^2} \mathbf{X}' \mathbf{y} - \frac{1}{\sigma_e^2} \left[\mathbf{X}' \mathbf{X} + \frac{\sigma_e^2}{\sigma_b^2} \mathbf{I} \right] \mathbf{X}' \mathbf{Z} \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix},$$

$$\frac{\partial f}{\partial b_i} \propto \frac{1}{\sigma_e^2} \mathbf{x}_i' \mathbf{y} - \frac{1}{\sigma_e^2} \left[\mathbf{x}_i' \mathbf{X}_{-i} \quad \mathbf{x}_i' \mathbf{x}_i + \frac{\sigma_e^2}{\sigma_b^2} \mathbf{I} \right] \mathbf{x}_i' \mathbf{Z} \begin{bmatrix} \mathbf{b}_{-i} \\ b_i \\ \mathbf{a} \end{bmatrix},$$

$$\frac{\partial f}{\partial \mathbf{a}} \propto \frac{1}{\sigma_e^2} \mathbf{Z}' \mathbf{y} - \frac{1}{\sigma_e^2} \left[\mathbf{Z}' \mathbf{X} \quad \mathbf{Z}' \mathbf{Z} + \frac{\sigma_e^2}{\sigma_a^2} \mathbf{A}^{-1} \right] \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix},$$

and

$$\frac{\partial f}{\partial a_i} \propto \frac{1}{\sigma_e^2} \mathbf{z}_i' \mathbf{y} - \frac{1}{\sigma_e^2} \left[\mathbf{z}_i' \mathbf{X} \quad \mathbf{z}_i' \mathbf{Z}_{-i} + \frac{\sigma_e^2}{\sigma_a^2} \mathbf{A}_{i-i}^{-1} \quad \mathbf{z}_i' \mathbf{z}_i + \frac{\sigma_e^2}{\sigma_a^2} \mathbf{A}_{ii}^{-1} \right] \begin{bmatrix} \mathbf{b} \\ \mathbf{a}_{-i} \\ a_i \end{bmatrix},$$

where $\mathbf{b}_{-i}$ is the vector of $\mathbf{b}$ without b_i , $\mathbf{a}_{-i}$ is the vector of $\mathbf{a}$ without a_i , $\mathbf{x}_i$ is the i th column vector relating to b_i , $\mathbf{X}_{-i}$ is the matrix relating to $\mathbf{b}_{-i}$, $\mathbf{z}_i$ is the i th column vector relating to a_i , and $\mathbf{Z}_{-i}$ is the matrix relating to $\mathbf{a}_{-i}$. $\mathbf{A}_{ii}^{-1}$ is the scalar value of the i th row and i th column of $\mathbf{A}^{-1}$, and $\mathbf{A}_{i-i}^{-1}$ is the i th row vector of $\mathbf{A}^{-1}$ without $\mathbf{A}_{ii}^{-1}$. In each equation, the first terms of the right hands are the right hands of the Henderson's mixed model equations, and the second terms are the left hands of the mixed model equations.

Appendix III

Appendix III demonstrates R codes for the HMC method. Table A1 shows a brief description of the variables we used in our R code.

Table A1. Variables used in our R codes

Basic parameters	num.p (constant)	a total size of fixed effect
	num.ped (constant)	the number of pedigrees
	n (constant)	the number of phenotypes
Model description	y (constant)	phenotypic vector (n)
	b	fixed effect work vector (num.p)
	u	random effect (breeding values) work vector (size num.ped)
	X (constant)	designed matrix for b (n, num.p)
	Z (constant)	designed matrix for u (n, num.ped)
	A.inv (constant)	a matrix of the inverse of an additive relationship matrix (num.ped, num.ped)
Variances	tau (constant)	prior variance for b, we set tau = 10,000
	var.u	genetic variance work variable
	var.e	residual variance work variable
Computing efficiency vector	xx (constant)	diagonal elements for $\mathbf{X}'\mathbf{X}$ (num.p)
	zz (constant)	diagonal elements for $\mathbf{Z}'\mathbf{Z}$ (num.ped)
Leapfrog integration	epsilon	ϵ
	L (constant to 7)	L
	max.L (constant to 20)	L_{one_round} ; maximum iterations per one round


```

569  1) R code for the HMC method of fixed effects
570  L <- 7; max.L <- 20
571
572  e <- y-X%*%b-Z%*%u
573  for(i in 1:num.p){
574    epsilon <- sqrt(1/(xx[ii]/var.e+1/tau))/(0.1589825*max.L)
575    b.tmp <- b[i]
576    xe <- crossprod(e+X[,i]*b.tmp, X[,i])
577    xx <- crossprod(X[,i])
578    p <- rnorm(1)
579    K0 <- t(p)%*%p/2
580    U0 <- -((2*xe*b.tmp-xx[i]*b.tmp^2)/(2*var.e)-b.tmp^2/(2*tau))
581    H0 <- (U0+K0)
582    for(t in 1:L){
583      p <- p-0.5*epsilon*(-((xe-xx[jj]*b.tmp)/var.e-b.tmp/tau))
584      b.tmp <- b.tmp+epsilon*p
585      p <- p-0.5*epsilon*(-((xe-xx[jj]*b.tmp)/var.e-b.tmp/tau))
586    }
587    K1 <- t(p)%*%p/2
588    U1 <- -((2*xe*b.tmp-xx[jj]*b.tmp^2)/(2*var.e)-b.tmp^2/(2*tau))
589    H1 <- (U1+K1)
590    if(runif(1) > exp(H0-H1)) b.tmp <- b[i]
591    e <- e+X[,i]*c(b[i]-b.tmp)
592    b[i] <- b.new
593  }

```

594

595 2) R code for the HMC method of breeding values

596 $e \leftarrow y - X\% \% b - Z\% \% u$

597 for(i in 1:num.ped){

598 $\epsilon \leftarrow \sqrt{1/(zz[jj]/var.e + A.inv[jj,jj]/var.u)} / (0.1589825 * max.L)$

599 $u.tmp \leftarrow u[i]$

600 $ze \leftarrow \text{crossprod}(e + Z[,i] * u.tmp, Z[,i])$

601 $zz \leftarrow \text{crossprod}(Z[,i])$

602 $uG \leftarrow \text{crossprod}(A.inv[,i], u) - A.inv[i,i] * u.tmp$

603 $p \leftarrow \text{rnorm}(1)$

604 $K0 \leftarrow t(p) \% \% p / 2$

605 $U0 \leftarrow -((2 * ze * u.tmp - zz[i] * u.tmp^2) / (2 * var.e) - (2 * uG * u.tmp + A.inv[i,i] * u.tmp^2) / (2 * var.u))$

606 $H0 \leftarrow (U0 + K0)$

607 for(t in 1:L){

608 $p \leftarrow p - 0.5 * \epsilon * (-((ze - zz[i] * u.tmp) / var.e - (uG + A.inv[i,i] * u.tmp) / var.u))$

609 $u.tmp \leftarrow u.tmp + \epsilon * p$

610 $p \leftarrow p - 0.5 * \epsilon * (-((ze - zz[i] * u.tmp) / var.e - (uG + A.inv[i,i] * u.tmp) / var.u))$

611 }

612 $K1 \leftarrow t(p) \% \% p / 2$

613 $U1 \leftarrow -((2 * ze * u.tmp - zz[i] * u.tmp^2) / (2 * var.e) - (2 * uG * u.tmp + A.inv[i,i] * u.tmp^2) / (2 * var.u))$

614 $H1 \leftarrow (U1 + K1)$

615 if(runif(1) > exp(H0 - H1)) $u.tmp \leftarrow u[i]$

616 $e \leftarrow e + Z[,i] * c(u[i] - u.tmp)$

617 $u[i] \leftarrow u.new$

618 }

619

620 **3) R code for the HMC method of genetic variance**

621 `var.u.tmp <- var.u`

622 `uAu <- t(u)%%a.inv%%u`

623 `epsilon <- sqrt(uAu**2/((num.ped-1)^2*(num.ped-2)))/(0.112485939* max.L)`

624 `p <- rnorm(1)`

625 `K0 <- t(p)%%p/2`

626 `U0 <- -((num.ped+v.u)/2+1)*log(var.u.tmp)-(uAu+lambda.u)/(2*var.u.tmp)`

627 `H0 <- (U0+K0)`

628 `for(t in 1:L){`

629 `p <- p-0.5*epsilon*(-((num.ped+v.u)/2+1)/var.u.new+`

630 `0.5*(uAu+lambda.u)/(var.u.tmp^2))`

631 `var.u.tmp <- var.u.tmp+epsilon*p`

632 `p <- p-0.5*epsilon*(-((num.ped+v.u)/2+1)/var.u.tmp+`

633 `0.5*(uAu+lambda.u)/(var.u.tmp^2))`

634 `}`

635 `K1 <- t(p)%%p/2`

636 `U1 <- -((num.ped+v.u)/2+1)*log(var.u.tmp)-(uAu+lambda.u)/(2*var.u.tmp)`

637 `H1 <- (U0+K0)`

638 `if(runif(1) < exp(-H1+H0)) var.u <- var.u.tmp`

639

640 **4) R code for the HMC method of residual variance**

641 `var.e.tmp <- var.e`

642 `e <- y-X%%b-Z%%u`

643 `ee <- t(e)%%e`

```

644  epsilon <- sqrt(ee**2/((n-1)^2*(n-2)))/(0.112485939* max.L)
645  p <- rnorm(1)
646  K0 <- t(p)%**%p/2
647  U0 <- -(-(n+v.e)/2+1)*log(var.e.tmp)-(ee+lambda.e)/(2*var.e.tmp))
648  H0 <- (U0+K0)
649  for(t in 1:L){
650    p <- p-0.5*epsilon*(-(-(n+v.e)/2+1)/var.e.tmp+
651      0.5*(ee+lambda.e)/(var.e.tmp^2)))
652    var.e.tmp <- var.e.tmp+epsilon*p
653    p <- p-0.5*epsilon*(-(-(n+v.e)/2+1)/var.e.tmp+
654      0.5*(ee+lambda.e)/(var.e.tmp^2)))
655  }
656  K1 <- t(p)%**%p/2
657  U1 <- -(-(n+v.e)/2+1)*log(var.e.new)-(ee+lambda.e)/(2*var.e.new))
658  H1 <- (U1+K1)
659  if(runif(1) < exp(H0-H1)) var.e <- var.e.new

```

References

- Aguilar, I., I. Misztal, D. J. Johnson, A. Legarra, S. Tsuruta, and T. J. Lawlor, 2010 Hot topic: a unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. *J Dairy Sci.*93: 743-752. doi:10.3168/jds.2009-2730.
- Amari, S., 2016 Information geometry and its applications. Springer, Japan.
- Betancourt M, S. Byrne, S. Livingstone, and M. Girolami, 2017 The geometric foundations of Hamiltonian monte carlo. *Bernoulli*. 23: 2257-2298. doi: 10.3150/16-BEJ810.
- Carpenter, B., A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, et al., 2017 Stan: A probabilistic programming language. *J. Stat. Soft.* 76: 1. doi:10.18637/jss.v076.i01.
- Cleveland, M. A., J. M. Hickey, and S. A. Forni, 2012 Common dataset for genomic analysis of livestock populations. *G3 (Bethesda)* 2: 429-435. doi:10.1534/g3.111.001453.
- Da, Y., C. Wang, S. Wang, and G. Hu, 2014 Mixed model methods for genomic prediction and variance component estimation of additive and dominance effects using SNP markers. *PLoS ONE* 9: e87666. doi:10.1371/journal.pone.0087666.
- Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth, 1987 Hybrid monte carlo. *Phys. Lett. B* 195: 216-222. doi:10.1016/0370-2693(87)91197-X.
- García-Cortés, L. A., and D. Sorensen, 1996 On a multivariate implementation of the Gibbs sampler. *Genet. Sel. Evol.* 28: 121-126. doi:10.1186/1297-9686-28-1-121.
- Girolami M, and B. Calderhead, 2011 Riemann manifold Langevin and Hamiltonian monte carlo methods. *J. Royal. Stat. Soc. Ser. B.* 73: 123-214. doi:10.1111/j.1467-9868.2010.00765.x.

- Habier, D., R. L. Fernando, K. Kizilkaya, D. J. Garrick, 2011 Extension of the Bayesian alphabet for genomic selection. BMC Bioinform. 12:186. <https://doi.org/10.1186/1471-2105-12-186>
- Hoffman, M. D., and A. Gelman, 2014 The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res. 15: 1593-1623.
- Ibáñez-Escriche, N., D. Sorensen, R. Waagepetersen, and A. Blasco, 2008 Selection for environmental variation: a statistical analysis and power calculations to detect response. Genetics 180: 2209-2226. doi:10.1534/genetics.108.091678.
- Jamrozik, J., and L. R. Schaeffer, 1997 Estimates of genetic parameters for a test day model with random regressions for yield traits of first lactation Holsteins. J Dairy Sci. 80: 762-770. doi:10.3168/jds.S0022-0302(97)75996-4.
- Jannink, J. L., A. J. Lorenz, and H. Iwata, 2010 Genomic selection in plant breeding: from theory to practice. Brief. Funct. Genom. 9: 166-177. doi:10.1093/bfpg/elq001.
- Meuwissen, T. H., B. J. Hayes, and M. E. Goddard, 2001 Prediction of total genetic value using genome-wide dense marker maps. Genetics 157: 1819-1829.
- Misztal, I., 2014 Computational techniques in animal breeding. Retrieved on 20 April 2016. http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=course16_uga.pdf
- Neal, R. M., 2011 MCMC using Hamiltonian dynamics, pp. p. 113-162 in: Handbook of Markov Chain Monte Carlo, edited by Gelman, S., A. Jones, and X. L. Meng. Chapman & Hall / CRC Press. doi:10.1201/b10905-6.
- Paquet, U., and M. Fraccaro, 2016 An efficient implementation of Riemannian manifold Hamiltonian Monte Carlo for Gaussian process models. arXiv Available at: <https://arxiv.org/abs/1810.11893>.
- Plummer, M., N. Best, K. Cowles, and K. Vines, 2006 CODA: convergence diagnosis and output analysis for MCMC. R News. 6: 7-11.

- Rao, C. R., 1945 Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.* 37: 81-91.
- Runcie, D. E., and S. Mukherjee, 2013 Dissecting high-dimensional phenotypes with Bayesian sparse factor analysis of genetic covariance matrices. *Genetics* 194: 753-767. doi:10.1534/genetics.113.151217.
- Sargolzaei, M., and F. S. Schenkel, 2009 QMSim: a large-scale genome simulator for livestock. *Bioinformatics*. 25: 680-81. doi:10.1093/bioinformatics/btp045.
- Sorensen, D. A., S. Andersen, D. Gianola, and I. Korsgaard, 1995 Bayesian inference in threshold models using Gibbs sampling. *Genet. Sel. Evol.* 27: 229-249. doi:10.1051/gse:19950303.
- Van Tassell, C. P., and L. D. Van Vleck, 1996 Multiple-trait Gibbs sampling for animal models: flexible programs for Bayesian and likelihood-based (co)variance component inference. *J. Anim. Sci.* 74: 2586-2597. doi:10.2527/1996.74112586x
- VanRaden, P. M., 2008 Efficient methods to compute genomic predictions. *J. Dairy Sci.* 91: 4414–4423. doi:10.3168/jds.2007-0980.
- Vitezica, Z. G., L. Varona, and A. Legarra, 2013 On the additive and dominant variance and covariance of individuals within the genomic selection scope. *Genetics* 195: 1223-1230. doi:10.1534/genetics.113.155176.
- Waldmann, P., J. Hallander, F. Hoti, and M. J. Sillanpää, 2008 Efficient Markov chain Monte Carlo implementation of Bayesian analysis of additive and dominance genetic variances in noninbred pedigrees. *Genetics* 179:1101-1112. doi:10.1534/genetics.107.084160.
- Wang, C. S., J. J. Rutledge, and D. Gianola, 1994 Bayesian analysis of mixed linear models via Gibbs sampling with an application to litter size in Iberian pigs. *Genet. Sel. Evol.* 26: 91-115. doi:10.1186/1297-9686-26-2-91.

732 Wang, C. S., J. J. Rutledge, D. Gianola, 1993 Marginal inferences about variance components
 733 in a mixed linear model using Gibbs sampling. Genet. Sel. Evol. 25: 41-62.
 734 doi:10.1186/1297-9686-25-1-41.

Figure legends

Figure 1. An example of a trajectory for Hamiltonian dynamics approximated by leapfrog integration. The horizontal axis shows the potential or random variable (θ), and the vertical axis shows the momentum variable (p). The stepsize is ϵ , and the number of step is L ($0 < t < L$). The initial state is at t_0 , and using leapfrog integration, θ and p are moved to the next state (t_1). The number of steps for one round of the trajectory is L_{one_round} , and the total length of the trajectory is expressed as ϵL_{one_round} .

Figure 2. Sampling states regarding the breeding value of the phenotyped individual having the highest effective sample size. (a) The trace plot between 5,000 and 5,500. (b) Autocorrelations after the burn-in period with a sampling lag of 1–40.

Figure 3. Marginal distributions of genetic (σ_a^2) and residual (σ_e^2) variances for t1 using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods. (a) σ_a^2 by HMC, (b) σ_a^2 by GS, (c) σ_e^2 by HMC, and (d) σ_e^2 by GS.

Figure 4. Marginal distributions of genetic (σ_a^2) and residual (σ_e^2) variances for t5 using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods. (a) σ_a^2 by HMC, (b) σ_a^2 by GS, (c) σ_e^2 by HMC, and (d) σ_e^2 by GS.

Figure 5. Marginal distributions of genomic (σ_g^2) and residual (σ_e^2) variances for t1 using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods. (a) σ_a^2 by HMC, (b) σ_a^2 by GS, (c) σ_e^2 by HMC, and (d) σ_e^2 by GS.

Figure 6. Marginal distributions of genomic (σ_g^2) and residual (σ_e^2) variances for t5 using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods. (a) σ_a^2 by HMC, (b) σ_a^2 by GS, (c) σ_e^2 by HMC, and (d) σ_e^2 by GS.

Figure 7. Marginal distributions of additive genomic (σ_g^2), dominance genomic (σ_d^2), and residual (σ_e^2) variances for t5 using the Hamiltonian Monte Carlo (HMC) and Gibbs sampling (GS) methods. (a) σ_g^2 by HMC, (b) σ_g^2 by GS, (c) σ_d^2 by HMC, (d) σ_d^2 by GS, (e) σ_e^2 by HMC, and (f) σ_e^2 by GS.

Tables

Table 3. Estimates by Hamiltonian Monte Carlo (HMC) with different L values under 20 iterations per round of the trajectory for leapfrog integration and Gibbs sampling (GS) methods

L	Residual Variance			Genetic Variance			Breeding Values			
	Estimate	Accept	ESS	Estimate	Accept	ESS	Cor	Slope	Accept ¹	ESS ¹
1	0.49 ± 0.06	9,969.4	23.6	0.52 ± 0.10	9,976.6	11.2	0.76	0.93	9,975.3 ± 5.0	169.3 ± 36.4
2	0.49 ± 0.05	9,948.6	53.0	0.53 ± 0.09	9,952.8	29.4	0.77	0.92	9,952.9 ± 6.8	443.8 ± 132.2
3	0.48 ± 0.06	9,934.2	85.4	0.54 ± 0.10	9,937.6	61.5	0.77	0.91	9,935.1 ± 8.1	345.4 ± 157.6
4	0.48 ± 0.05	9,920.4	147.1	0.53 ± 0.10	9,924.2	111.6	0.77	0.92	9,924.0 ± 8.8	470.2 ± 312.2
5	0.49 ± 0.05	9,923.6	289.5	0.53 ± 0.09	9,923.2	215.3	0.77	0.92	9,920.3 ± 8.8	614.7 ± 597.1
6	0.48 ± 0.06	9,918.2	417.7	0.53 ± 0.10	9,925.4	319.1	0.77	0.92	9,924.5 ± 8.7	731.0 ± 1130.7
7	0.48 ± 0.06	9,933.2	716.8	0.53 ± 0.10	9,938.8	473.7	0.77	0.92	9,936.1 ± 7.9	1,215.8 ± 2,163.4
8	0.48 ± 0.06	9,955.8	750.8	0.53 ± 0.10	9,949.4	370.6	0.77	0.92	9,954.0 ± 6.9	2,164.8 ± 4,032.5
9	0.48 ± 0.05	9,968.8	333.4	0.54 ± 0.10	9,968.4	125.0	0.77	0.91	9,976.4 ± 4.8	7,482.8 ± 12,514.6
10	0.09 ± 0.01	9,975.6	10.0	1.39 ± 0.11	9,978.4	10.0	0.70	0.53	9,998.7 ± 1.2	67,267.7 ± 237,508.4
11	0.48 ± 0.06	9,966.6	410.9	0.54 ± 0.10	9,963.2	163.5	0.77	0.91	9,974.0 ± 5.1	7,025.2 ± 9,861.3
12	0.48 ± 0.05	9,958.6	787.4	0.53 ± 0.10	9,948.0	395.0	0.77	0.92	9,951.8 ± 7.0	2,121.0 ± 3,853.8
13	0.48 ± 0.06	9,944.4	633.8	0.53 ± 0.10	9,940.0	397.8	0.77	0.92	9,934.5 ± 8.1	1,193.1 ± 1,982.1
14	0.48 ± 0.06	9,925.0	407.2	0.54 ± 0.10	9,919.4	305.2	0.77	0.92	9,923.4 ± 8.8	715.6 ± 1,061.7
15	0.48 ± 0.06	9,917.4	244.1	0.54 ± 0.10	9,917.4	188.7	0.77	0.91	9,920.2 ± 8.9	605.2 ± 565.6

16	0.48 ± 0.06	9,928.0	141.0	0.53 ± 0.10	9,923.6	107.1	0.77	0.92	9,924.9 ± 8.6	458.7 ± 297.2
17	0.49 ± 0.06	9,937.6	76.4	0.52 ± 0.10	9,933.4	51.0	0.77	0.92	9,936.8 ± 7.9	581.7 ± 198.6
18	0.48 ± 0.06	9,958.0	44.8	0.53 ± 0.10	9,959.8	24.9	0.77	0.92	9,955.2 ± 6.6	460.7 ± 134.7
19	0.56 ± 0.10	9,980.6	8.1	0.40 ± 0.15	9,979.2	3.8	0.76	1.06	9,977.8 ± 4.8	121.9 ± 33.1
20	1.06 ± 0.06	9,997.4	2.8	0.02 ± 0.00	9,995.8	4.0	0.15	1.43	9,997.3 ± 1.8	5.2 ± 2.7
GS	0.48 ± 0.05		233.2	0.53 ± 0.09		188.0	0.77	0.92		564.7 ± 600.0

773 [†]Accept and ESS were averaged for all animals with a phenotypic record.

774 ESS, effective sample size

Figure 1.

Figure 2.

(a) Genetic variance by HMC

(b) Genetic variance by GS

(c) Residual variance by HMC

(d) Residual variance by GS

Figure 3. trait 1 pedigree

(a) Genetic variance by HMC

(b) Genetic variance by GS

(c) Residual variance by HMC

(d) Residual variance by GS

Figure 4. trait 5 pedigree

(a) Genetic variance by HMC

(b) Genetic variance by GS

(c) Residual variance by HMC

(d) Residual variance by GS

Figure 5. trait 1 genomic

Figure 6. trait 5 genomic

Figure 7.