Modelling correlated marker effects in genome-wide prediction via Gaussian concentration graph models

Carlos Alberto Martínez[1], Kshitij Khare[2], Syed Rahman[2], Mauricio A. Elzo[1]

[1]Department of Animal Sciences

[2]Department of Statistics

University of Florida, Gainesville, FL, USA

Correspondence: Carlos Alberto Martínez, Department of Animal Sciences, University of Florida, Gainesville, FL 32611, USA.

Tel: 352-328-1624.

Fax: 352-392-7851.

E-mail: carlosmn@ufl.edu

ABSTRACT

In genome-wide prediction, independence of marker allele substitution effects is typically assumed; however, since early stages in the evolution of this technology it has been known that nature points to correlated effects. In statistics, graphical models have been identified as a useful and powerful tool for covariance estimation in high dimensional problems and it is an area that has recently experienced a great expansion. In particular, Gaussian concentration graph models (GCGM) have been widely studied. These are models in which the distribution of a set of random variables, the marker effects in this case, is assumed to be Markov with respect to an undirected graph G. In this paper, Bayesian (Bayes G and Bayes G-D) and frequentist (GML-BLUP) methods adapting the theory of GCGM to genome-wide prediction were developed. Different approaches to define the graph G based on domain-specific knowledge were proposed, and two propositions and a corollary establishing conditions to find decomposable graphs were proven. These methods were implemented in small simulated and real datasets. In our simulations, scenarios where correlations among allelic substitution effects were expected to arise due to various causes were considered, and graphs were defined on the basis of physical marker positions. Results showed improvements in correlation between phenotypes and predicted additive genetic values and accuracies of predicted additive genetic values when accounting for partially correlated allele substitution effects. Extensions to the multiallelic loci case were described and some possible refinements incorporating more flexible priors in the Bayesian setting were discussed. Our models are promising because they allow incorporation of biological information in the prediction process, and because they are more flexible and general than other models accounting for correlated marker effects that have been proposed previously.

Key words Correlated allele substitution effects, genome-enabled prediction, graphical models, sparse covariance estimation.

1. INTRODUCTION

A feature shared by different statistical models used in genome-wide prediction (Meuwissen et al., 2001) like the family of hierarchical Bayesian regression models known as the Bayesian alphabet (Gianola et al., 2009; Gianola, 2013) and methods based on mixed model equations known as G-BLUP (VanRaden, 2008) and single step G-BLUP (Aguilar et al., 2011), is that marker allele substitution effects are assumed to be mutually independent. Therefore, a diagonal structure for the covariance matrix of these effects is always assumed. However, it is well known that in many populations, especially livestock and plant populations that have undergone selection, linkage disequilibrium (LD) exists. The presence of LD combined with the interactions among genes and complex interactions between gene products taking place in the metabolism point to correlated marker effects. For example, when several SNP are in LD with the same QTL or group of QTL, their effects could be correlated. Another case where SNP effects are expected to be correlated is when a set of SNP are linked to some QTL whose products interact along metabolic pathways.

Thus, there is a need to account for the correlation among marker effects because this is closer to reality and brings advantages like the use of information from correlated marker effects in the prediction of the effect of a given marker, inclusion of biological information in the prediction process and a better knowledge of the covariance structure of marker effects for a particular trait or set of traits. So far, there have been few studies attempting to model a non-diagonal covariance matrix of marker effects in genome-wide prediction and none of them has used an approach based on graphical models. Gianola et al. (2003) proposed Bayesian and frequentist approaches to account for correlated SNP allele substitution effects whereas Yang and Tempelman (2012) developed a hierarchical Bayesian model imposing an autoregressive structure on the covariance matrix of these effects.

In statistics, estimation of large covariance matrices when the number of variables is larger than the sample size ("small n, large p") is an important problem of current interest (Khare and Rajaratnam, 2011; Oh et al., 2016). A growing research area to deal with this complex problem is the

use of graphical models to find regularized estimators of covariance or precision matrices using both frequentist and Bayesian methods (Carvalho et al., 2007; Letac and Massan, 2007). However, the idea of imposing zeros in the precision matrix is not new; it was proposed by Dempster (1972). Many of these methods share the underlying property of yielding shrinkage estimators (Rajaratnam et al., 2008). Graphical models induce zeros in the covariance or precision matrix, thereby reducing the total number of parameters to be estimated. Models inducing sparsity in the covariance matrix are called covariance graph models, while those inducing sparsity in the precision matrix are called concentration graph models. When the joint distribution of the underlying variables (marker effects in this case) is assumed to be multivariate Gaussian, these zeros translate to appropriate marginal independence assumptions (Gaussian covariance graph models) or conditional independence assumptions (Gaussian concentration graph models).

The patterns of zeros in the covariance or precision matrix can be encoded in terms of an undirected graph G, hence the term "graphical models". (Dempster, 1972; Dawid and Lauritzen, 1993; Silva and Ghahramani, 2006; Khare and Rajaratnam, 2011; Ben-David et al., 2015). The nodes of this undirected graph G represent the underlying random variables. For Gaussian concentration graph models (GCGM), variables not sharing an edge in G are conditionally independent given all other variables (Dawid and Lauritzen, 1993). Equivalently, some authors refer to this property as the undirected Markov property with respect to G (Ben-David et al., 2015). Covariance estimation using graph models is an appealing and very promising topic in statistical genomics because it allows accounting for correlation of marker effects in the prediction of genotypic values and phenotypes.

Applications of graphical models in genome-wide prediction have been focused on directed acyclic graph models (AKA Bayesian networks). Some studies applying this kind of graphical models to prediction problems in quantitative genomics are Malovini et al. (2009), Chang and McGeachie (2011), Han et al. (2012), Scutari et al. (2013), and Scutari et al. (2014). To our knowledge, we are the

4

first to adapt GCGM to account for correlated SNP allele substitution effects in genome-wide prediction. A challenge encountered in this process is the following. The theory of GCGM is developed to estimate the precision matrix of an observable p-dimensional random vector using a sample of size n. However, estimation of dispersion parameters in genome-wide prediction involve the problem of estimating residual variance(s) and the more challenging problem of estimating the covariance or precision matrix of an unobservable random vector containing SNP effects using a single n-dimensional vector of phenotypes as well as genotypes. Thus, the objectives of this study were to introduce the theory of GCGM in genome-wide prediction by developing methods to adapt it to perform sparse estimation of the precision matrix of allele substitution effects, and to evaluate its impact on the accuracy of genome-wide prediction.

2. METHODS

This section is organized as follows. Because the theory of GCGM is not widely known by quantitative geneticists, it is briefly presented in subsection 2.1 along with the challenges encountered when adapting it to genome-wide prediction. Subsection 2.2 presents the Bayesian approach to solve the problem (Bayes G and Bayes G-D), whereas an EM algorithm is developed in subsection 2.3 to provide a frequentist solution (GML-BLUP). In subsection 2.4, several approaches to build the graph G, some of them leading to decomposable graphs, are presented. Finally, in subsection 2.5, simulated and real datasets used to implement our methods are described.

2.1 Gaussian Concentration Graph Models

In this section, frequentist and Bayesian approaches for Gaussian concentration graph models are presented. The off-diagonal entries of the inverse covariance matrix correspond to the conditional covariance between pairs of variables given all other variables; therefore, under a GCGM, assumptions about the structure of this matrix are made and, as explained in the introduction, this structure is represented using a graph G (Dawid and Lauritzen, 1993; Letac and Massam, 2007).

When the pattern of zeros is unknown, i.e., the graph is unknown; the interest is to find the null entries and to estimate the non-null entries of the inverse covariance matrix. This problem is known as model selection (Bickel and Levina 2008; Rajaratnam et al., 2008; Khare et al., 2015). Here, the case of a known graph G is considered. The scenario of known G encompasses situations where either the pattern of zeros of the precision matrix is actually known or domain-specific knowledge permits the definition of G. Before discussing concentration graph models, the reader not familiar with basic concepts in graph theory is referred to Appendix A.

2.1.1 The estimation problem

The statistical problem is the following. Suppose that $Y_1, Y_2, \ldots, Y_n$ is a set of vectors in $\mathbb{R}^p$ identically and independently distributed $MVN(0, \Sigma)$. The inverse of the covariance matrix Σ is usually denoted by Ω. In a GCGM, the target is to estimate Ω instead of Σ. Let $G = (V, E)$ be a graph with vertices set V and edges set E, notice that $|V| = p$. As mentioned before, the graph G defines the null entries in Ω and hence it defines the sparsity pattern. The vertices of G represent the set of random variables we are dealing with. Formally, the parameter space is the following cone (Dawid and Lauritzen, 1993): $\Omega \in \mathbb{P}_G = \{A : A \in \mathbb{P}^+ \text{ and } A_{ij} = 0 \text{ whenever } (i,j) \notin E\}$, where $\mathbb{P}^+$ is the space of positive definite matrices.

Maximum likelihood estimation

The negative of the log-likelihood function has the form:

$$l(\Omega) = c + \frac{n}{2} tr(\Omega S) - \frac{n}{2} log|\Omega|, \Omega \in \mathbb{P}_G,$$

where c involves all terms not depending on Ω, and S is the sample covariance matrix defined as:

$$S = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \overline{Y})(Y_i - \overline{Y})', \overline{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i.$$

The negative of the log-likelihood can be slightly modified to obtain the following objective function whose minimization is equivalent to the minimization of $l(\Omega)$:

$$l^*(\Omega) = tr(\Omega S) - log|\Omega|, \Omega \in \mathbb{P}_G \qquad (2.1)$$

In general, there is no closed form solution. An iterative proportional fitting algorithm developed by Speed and Kiiveri (1986) allows minimizing $l^*(\Omega)$. It is based on a partition of the covariance matrix according to the maximal cliques of G.

The properties of the graph have mathematical and statistical consequences on the estimation problem (Letac and Massan, 2007; Khare and Rajaratnam, 2011, Khare and Rajaratnam, 2012). Therefore, it is important to assess certain properties of G because their impact on the estimation problem can be advantageous. In this case, it turns out that when n is greater than the size of the largest clique in G, $l(\Omega)$ is strictly convex and consequently it has a unique global minimum. In addition, for decomposable graphs, the MLE has a closed form. Now, some results needed to derive this closed form and are presented; these will also be used in the Bayesian estimation of Ω.

If Ω is a positive definite matrix, there exists a pair (L, D) such that: $\Omega = LDL'$, L is lower triangular with diagonals equal to one, D is positive diagonal (i.e., $D_{ii} > 0 \ \forall \ i = 1,2, \dots, p$), and the pair (L, D) is unique. This decomposition is known as modified Cholesky decomposition. Decomposable graphs satisfy the following property (Paulsen et al., 1989). The graph $G = (V, E)$ is decomposable, if and only if, for every $\Omega \in \mathbb{P}_G$ with modified Cholesky decomposition $\Omega = LDL'$ it follows that $L \in \mathcal{L}_G = \{L: L_{ii} = 1, L_{ij} = 0 \ whenever \ i < j \ or \ (i, j) \notin E, \forall \ 1 \leq i, j \leq p\}$. The implication of this result is that the Cholesky matrix L has (in its lower diagonal elements) the same pattern of zeros as Ω. Now consider the following space: $\mathcal{D} = \{D: D \ is \ diagonal \ with \ D_{ii} > 0 \ \forall \ i = 1,2, \dots, p\}$. To come up with the closed form for the MLE of Ω when G is decomposable, the minimization is done in terms of the modified Cholesky decomposition of Ω, that is, we work with the transformation $\Omega \rightarrow (L, D)$ that corresponds to a bijection from $\mathbb{P}_G$ to $\mathcal{L}_G \times \mathcal{D}$. Thus, the minimization is done with respect to (L, D). After some algebra, the objective function can be written as:

$$l^*(L, D) = \sum_{i=1}^{p} [D_{ii}(L_{.i}'SL_{.i}) - \log D_{ii}]$$

where $L_{.i}$ is the ith column of L. Then, focusing on the ith minimization problem in the above sum and taking into account the structure of $L_{.i}$ we can write:

$$L_{.i}'SL_{.i} = (1 \quad \boldsymbol{x}_i) \begin{pmatrix} S_{ii} & (\boldsymbol{S}_{.i}^{>})' \\ \boldsymbol{S}_{.i}^{>} & S^{>i} \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{x}_i \end{pmatrix} \tag{2.2}$$

where $\boldsymbol{x}_i$ is a vector of dimension v_i containing the non-constrained entries of $L_{.i}$ and it is formally defined as $\boldsymbol{x}_i = (L_{ji})_{j>i,(i,j)\in E}$, $S^{>i}$ is a submatrix of the sample covariance matrix built using the vertices greater than i that are connected to vertex i and it is defined as $S^{>i} = (S_{jk})_{j,k>i;(i,j),(i,k)\in E}$, $\boldsymbol{S}_{.i}^{>}$ is a column vector defined as $\boldsymbol{S}_{.i}^{>} = (S_{ji})_{j>i,(i,j)\in E}$. Hence, the function to be minimized for the ith case is:

$$D_{ii}(1 \quad \boldsymbol{x}_i) \begin{pmatrix} S_{ii} & (\boldsymbol{S}_{.i}^{>})' \\ \boldsymbol{S}_{.i}^{>} & S^{>i} \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{x}_i \end{pmatrix} - \log D_{ii}, D_{ii} > 0, \boldsymbol{x}_i \in \mathbb{R}^{v_i}.$$

After minimizing with respect to $\boldsymbol{x}_i$, plugging the solution into the objective function and minimizing with respect to D_{ii} we obtain:

$$\hat{\boldsymbol{x}}_i = (S^{>i})^{-1} \boldsymbol{S}_{.i}^{>} \tag{2.3}$$

$$\hat{D}_{ii} = \frac{1}{S_{ii}(\boldsymbol{S}_{.i}^{>})'(S^{>i})^{-1}\boldsymbol{S}_{.i}^{>}} \tag{2.4}$$

By combining $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{p-1}$ and $\{\hat{D}_{ii}\}_{i=1}^{p}$ matrices $\hat{L}$ and $\hat{D}$ can be built and by the invariance property of the MLE (Lehmann and Casella, 1998), the MLE of Ω can be written as $\hat{\Omega} = \hat{L}\hat{D}\hat{L}'$.

Bayesian estimation

In the Bayesian estimation of Ω, a family of priors commonly used is known as the Diaconis-Ylvisaker prior (Diaconis and Ylvisaker, 1979). They are appealing because among others, they have the property of yielding linear posterior expectations. The way to construct these priors comes from

the following theorem (Diaconis and Ylvisaker, 1979) which is stated here for the sake of completeness.

Consider a natural exponential family

$$f_\theta(x) = \exp(x'\theta - k(\theta)) h(x)$$

and a family of priors with a kernel of the form:

$$\tilde{\pi}_{n_0,x_0}(\theta) = \exp(n_0 x_0'\theta - n_0 k(\theta)),$$

now define the collection $\{\tilde{\pi}_{n_0,x_0}\}_{n_0,x_0}$ with η_0, x_0 such that $\tilde{\pi}_{n_0,x_0}(\theta)$ can be normalized. Under this sort of priors the following properties hold:

1. The posterior has the form $\tilde{\pi}_{n_0+n,\frac{n_0 x_0+n\bar{x}}{n_0+n}}(\theta)$

2. $E\left[\frac{\partial K(\theta)}{\partial \theta}\middle| X_1, X_2, \dots, X_n\right] = \frac{n_0 x_0 + n\bar{x}}{n_0+n}$

3. If $\pi(\theta)$ is any prior such that it is not concentrated on a single point and $E\left[\frac{\partial K(\theta)}{\partial \theta}\middle| X\right] = aX + b$ for some constants a and b, then $a \neq 0$ and $\pi(\theta)$ is of the form $ce^{\frac{1}{a}b\theta - \frac{1}{a}(1-a)k(\theta)}$, that is, $\pi(\theta)$ has to be a Diaconis-Ylvisaker prior.

The parameter x_0 can be interpreted as the mean of n_0 imaginary prior observations. Consequently, property 2 indicates that the posterior expectation is a convex combination of the prior and the sample mean. Finally, property 3 states posterior linearity. The general idea when building a Diaconis-Ylvisaker prior is to "imitate the likelihood". Recall that the problem of interest is to estimate the inverse covariance matrix Ω under multivariate normality, then the likelihood of Ω is:

$$L(\Omega) =\propto \exp\left(-\frac{(tr(\Omega nS) - n\log|\Omega|)}{2}\right), \Omega \in \mathbb{P}_G,$$

Even though $\Omega \in \mathbb{P}_G$, it can be proven that this is an appropriate density function of the unconstrained (non-zero) elements of Ω. When applying the "imitate the likelihood" principle to this problem, it can be shown that the Diaconis-Ylvisaker prior is:

$$\pi_{(U,\delta)}(\Omega) \propto |\Omega|^{\delta/2} \exp(-tr(\Omega U)/2), U \in \mathbb{P}^+, \delta > 0, \Omega \in \mathbb{P}_G.$$

The shape parameter δ can be also interpreted as a shrinkage parameter (Rajaratnam et al., 2008). Notice that, if $\Omega \in \mathbb{P}^+$ then we have an inverse Wishart density. However, the matrix Ω is restricted to $\mathbb{P}_G$ and because of that, this family of priors is known as the G-Wishart (GW) (Roverato, 2000). Under this prior and the likelihood $L(\Omega)$, by the Diaconis-Ylvisaker theorem it follows that the posterior density of Ω is a GW with parameters $nS + U$ and $n + \delta$, where S is the sample covariance matrix. As in the case of the MLE, it turns out that decomposable graphs have an appealing property. For decomposable graphs, posterior expectations can be obtained in closed form (Letac and Massam, 2007, Rajaratnam et al., 2008) and direct samples from the posterior distribution can be obtained, while for non-decomposable graphs MCMC methods can be used.

In the case of decomposable graphs, in a similar approach to the one used to derive the MLE, instead of working with Ω, its modified Cholesky decomposition $\Omega = LDL'$ is used. As will be shown, this allows direct sampling from the GW distribution. Recall that if the graph G is decomposable then $L \in \mathcal{L}_G$ and $D \in \mathcal{D}$. We already defined the density $\pi_{(U,\delta)}(\Omega)$, but now we are considering a bijection from $\mathbb{P}_G$ to $\mathcal{L}_G \times \mathcal{D}$; therefore, applying standard theory we can compute the density of L and D. To this end, the Jacobian of the transformation has to be found. To ease this computation, an ordering strategy of the entries of Ω is used. A brief description of its derivation is presented, for further details see Roverato (2002). The elements of L and D are ordered as follows:

$$D_1, L_{21}, D_2, L_{31}, L_{32}, \dots, D_j, L_{(j+1)1}, L_{(j+1)2}, \dots, L_{(j+1)j}, D_{j+1}, \dots, D_p,$$

then, the zero entries are ignored. On the other hand, the elements of Ω are ordered row-wise removing redundancies, that is, $\Omega_{11}, \Omega_{21}, \Omega_{22}, \dots, \Omega_{p(p-1)}, \Omega_{pp}$ and the zero entries are ignored as well. Recall that the null entries in the lower off diagonal part of L are the same of Ω. With this ordering strategy, it turns out that the Jacobian is an upper triangular matrix whose diagonal entries are either 1's or the D_{ii} such that $i \in \{i : i > j, (i,j) \in E\} \ \forall j = 1, 2, \dots, p - 1$. Thus, the determinant of the Jacobian is:

$$|J| = \prod_{j=1}^{p-1} D_j^{n_j},$$

where $n_j = |\{i : i > j, (i,j) \in E\}|$. The operator $|\cdot|$ represents determinant when the argument is a matrix and cardinality when the argument is a set. Therefore, the density on $\mathcal{L}_G \times \mathcal{D}$ induced by $\pi_{(U,\delta)}(\Omega)$ is:

$$\pi_{(U,\delta)}(L, D) \propto \exp\left(\frac{-tr(\Omega U)}{2}\right) \left(\prod_{j=1}^{p-1} D_j^{\frac{\delta + 2n_j}{2}}\right) D_p^{\frac{\delta}{2}}.$$

After some algebraic manipulations, this density can be rewritten as:

$$\pi_{(U,\delta)}(L, D) \propto \left(\prod_{j=1}^{p-1} \exp\left(-\frac{c_j D_j}{2}\right) D_j^{(\delta + 2n_j)/2} \exp\left(-\frac{D_j}{2}(x_j - a_j)'(U^{>j})(x_j - a_j)\right)\right)$$

$$\times \exp\left(\frac{-U_{pp} D_p}{2}\right) D_p^{\delta/2}$$

where x_j is defined as before, $a_j = -(U^{>j})^{-1} U_{\cdot j}^>$, $c_j = U_{jj} - (U_{\cdot j}^>)'(U^{>j})^{-1} U_{\cdot j}^>$, U_{jj} and $U^{>j}$ are the elements resulting from partitioning U in the same way that S was partitioned in (2.2). From $\pi_{(U,\delta)}(L, D)$ it follows that $\{(x_j, D_j)\}_{j=1}^{p-1}$ and D_p are mutually independent random variables, a fact that is advantageous when sampling from this joint distribution. Notice that:

$$\pi_{(U,\delta)}(x_j, D_j) \propto \exp\left(\frac{-c_j D_j}{2}\right) D_j^{\frac{\delta + 2n_j}{2}} \exp\left(-\frac{D_j}{2}(x_j - a_j)'(U^{>j})(x_j - a_j)\right), j = 1, 2, \dots, p-1$$

conditioning on D_j yields: $\pi(x_j | D_j) \propto \exp\left(-\frac{D_j}{2}(x_j - a_j)'(U^{>j})(x_j - a_j)\right)$ hence,

$x_j | D_j \sim MVN\left(a_j, \frac{1}{D_j}(U^{>j})^{-1}\right)$. Also:

$$\pi(D_j) \propto \int_{\mathbb{R}^{\nu_j}} \exp(-c_j D_j / 2) D_j^{(\delta + 2n_j)/2} \exp\left(-\frac{D_j}{2}(x_j - a_j)'(U^{>j})(x_j - a_j)\right) dx_j$$

$$\propto \exp(-c_j D_j / 2) D_j^{(\delta + n_j)/2},$$

11

consequently, $D_j \sim Gamma\left(\frac{\delta+n_j}{2} + 1, \frac{c_j}{2}\right)$. Similarly,

$$\pi_{(U,\delta)}(D_p) \propto \exp\left(-\frac{U_{pp}D_p}{2}\right) D_p^{\frac{\delta}{2}} \therefore D_p \sim Gamma\left(\frac{\delta}{2} + 1, \frac{U_{pp}}{2}\right).$$

Therefore, $\pi(x_j|D_j)$ and $\pi(D_j)$ are standard distributions and direct sampling can be implemented.

When the graph is non-decomposable, indirect approaches as MCMC methods can be used. One approach is the block-Gibbs sampler proposed by Piccioni (2000) which involves a partition of Ω according to the set of maximal cliques of the graph G. Although this partition is not proper in the sense that it is not disjoint, Piccioni (2000) showed that despite this fact, convergence of the algorithm is attained. Other methods have also being proposed to sample from a GW distribution. Mitsakakis et al. (2011) derived an independent Metropolis algorithm and Lenkoski (2013) came up with a method to obtain direct samples.

2.1.2 Challenges encountered when adapting GCGM to genome-wide prediction

In this study we approach genome-wide prediction as a multiple linear regression problem. The model considered here is as follows:

$$y = Wg + e \tag{2.5}$$

where y is an observable n-dimensional random vector containing response variables (e.g., corrected phenotypes), g is an unknown m-dimensional random vector of marker allele substitution effects, e is a vector of residuals, W is an observable random matrix containing one to one mappings from the set of possible genotypes for every individual at every locus to a subset of the integers which correspond to functions of counts of the reference allele. In this study, these mappings are defined as follows:

$$W = \{w_{ij}\} = \begin{cases} 1, if\ genotype = BB \\ 0, if\ genotype = BA \\ -1, if\ genotype = AA \end{cases}$$

where w_{ij} is the mapping of the genotype of the ith individual for the jth marker. The distributional assumptions are the following:

$$\boldsymbol{g}|\Omega\sim MVN(0,\Omega^{-1})$$

$$\boldsymbol{e}|\sigma^2\sim MVN(0,\sigma^2 I).$$

Under GCGM, the current methods require an independent and identically distributed (IID) sample which is assumed to be drawn from a multivariate normal distribution. The objective is not to estimate the covariance matrix of $\boldsymbol{y}$ (denoted as V) but Ω. On the other hand, residuals are commonly assumed to be homoscedastic and independent; therefore, $Var[\boldsymbol{e}]$ is a sparse matrix. Sometimes this matrix is assumed to have another structure, but it is generally sparse. Statistical methods for covariance estimation using graphical models have been proposed for the situation in which $V = L + S$ where L is a low rank symmetric matrix and S is a sparse matrix (Zhang et al., 2013). However, these methods require a sample $\boldsymbol{y_1},\boldsymbol{y_2},\dots,\boldsymbol{y_k}$ of n-dimensional vectors that in this particular case is not available because individuals are typically measured only once for a given trait. For example, the total milk yield of a cow in a given lactation and the weaning weight of a calf are measured only once. In addition, this methodology uses graphical models only to estimate the residual covariance or inverse covariance matrix, not the covariance matrix of regression coefficients (Zhang et al., 2013). Consequently, the theory of GCGM cannot be directly applied to genome-wide prediction and statistical methods have to be developed to adapt it to estimate Ω and predict marker effects under model 2.5. In the following, Bayesian and frequentist solutions to this problem are presented.

2.2 Bayesian approach

Hierarchical Bayesian models suit this problem very well because they provide a principled, natural, and simple way to estimate the inverse covariance matrix of the unobservable vector containing marker allele substitution effects. Given a graph G, the following layers are added to complete the hierarchical representation of model 2.5 which is named Bayes G:

$$\sigma^2\sim Inverse\ Gamma\left(\frac{\tau^2}{2},\frac{v}{2}\right)\coloneqq IG\left(\frac{\tau^2}{2},\frac{v}{2}\right)$$

$$\Omega|G\sim GW(\delta,U).$$

13

Under this prior, the joint posterior is:

$$\pi(\boldsymbol{g}, \sigma^2, \Omega | \boldsymbol{y}, G) \propto f(\boldsymbol{y} | \boldsymbol{g}, \sigma^2, W) \pi(\boldsymbol{g} | \Omega) \pi(\Omega | G) \pi(\sigma^2)$$

$$\propto (\sigma^2)^{-\frac{n}{2}} \exp\left(\frac{-1}{2\sigma^2}(\boldsymbol{y} - W\boldsymbol{g})'(\boldsymbol{y} - W\boldsymbol{g})\right)$$

$$\times |\Omega|^{1/2} \exp\left(\frac{-1}{2}\boldsymbol{g}'\Omega\boldsymbol{g}\right) |\Omega|^{\delta/2} \exp(-tr(\Omega U)/2)$$

$$\times (\sigma^2)^{-\left(\frac{v}{2}+1\right)} \exp\left(\frac{-\tau^2}{2\sigma^2}\right)$$

As mentioned before, for decomposable graphs, there exists a closed form for the posterior mean under the conventional GCGM. This is possible because the posterior distribution of Ω is known to be a $GW(\delta + n, U + nS)$, but under the model considered here, this posterior distribution cannot be found in closed form. Therefore, even for decomposable graphs, it is necessary to use an MCMC algorithm to sample from the posterior distribution in order to obtain point estimates of the precision matrix, the residual variance, and the marker effects. To implement a Gibbs sampler, all the full conditionals have to be known so that we can easily sample from them. All the distributions considered here are routinely used in regression models except for the prior for Ω; fortunately, due to the fact that the full conditional density of Ω under model 2.5 is $\pi(\Omega | Else) = \pi(\Omega | \boldsymbol{g}, G)$ and property 1 of the Diaconis-Ylvisaker prior shown above, it follows that $\Omega | Else \sim GW(\delta + 1, U + \boldsymbol{g}\boldsymbol{g}'), \Omega \in \mathbb{P}_G$, and consequently samples from this full conditional can be obtained using some of the methods mentioned before. This simple but key property allows a straightforward implementation of a Gibbs sampler in Bayes G because the remaining full conditionals are standard; these are shown for completeness.

$$\boldsymbol{g} | Else \sim MVN\left(\left(\Omega + \frac{W'W}{\sigma^2}\right)^{-1}\frac{1}{\sigma^2}W'\boldsymbol{y}, \left(\Omega + \frac{W'W}{\sigma^2}\right)^{-1}\right)$$

$$\sigma^2 | Else \sim IG\left(\frac{v + n}{2}, \frac{(\boldsymbol{y} - W\boldsymbol{g})'(\boldsymbol{y} - W\boldsymbol{g}) + \tau^2}{2}\right)$$

For general graphs, some of the methods developed to sample from the GW distribution are computationally demanding, and the time to obtain a single sample of the precision matrix of vectors of moderate size (e.g., 1000 to 2000) can make the implementation of a Gibbs sampler computationally intractable. Therefore, it is useful to take advantage of the properties of decomposable graphs whenever possible. Before focusing on the problem of finding decomposable graphs, which is deferred to subsection 2.4, the full conditional distributions of the Cholesky parameters of Ω given that G is decomposable are presented. Using the forms of the distributions of $x_j|D_j, j = 1,2, \dots, m-1$ and $D_j, j = 1,2, \dots, m$, presented in subsection 2.1.1, it follows that under model 2.5 $x_1, \dots, x_{m-1}|Else$ distribute independently as:

$$x_j|Else \sim MVN\left((U^{>j} + (gg')^{>j})^{-1}(U_{.j}^{>} + (gg')_{.j}^{>}), (U^{>j} + (gg')^{>j})^{-1}\right)$$

where matrices U and gg' are partitioned as in 2.2. Furthermore $D_1, \dots, D_m|Else$ distribute independently as:

$$D_j|Else \sim Gamma\left(\frac{\delta + n + n_j}{2} + 1, \frac{(gg')_{jj} + U_{jj} - Q(U,g)}{2}\right), j = 1,2, \dots, m-1$$

$$Q(U,g) = \left(U_{.j}^{>} + (gg')_{.j}^{>}\right)'(U^{>j} + (gg')^{>j})^{-1}\left(U_{.j}^{>} + (gg')_{.j}^{>}\right)$$

$$D_m|Else \sim Gamma\left(\frac{\delta + n}{2} + 1, \frac{(gg')_{mm} + U_{mm}}{2}\right).$$

This particular case of our hierarchical Bayesian model arising when G is decomposable is referred to as Bayes G-D. Sampling from these distributions at iteration t permits to build matrices $L^{(t)}$ and $D^{(t)}$ which in turn allow obtaining the precision matrix as $\Omega^{(t)} = L^{(t)}D^{(t)}\left(L^{(t)}\right)'$.

2.3 Frequentist approach

Unlike the Bayesian approach, envisaging a frequentist solution to the problem of adapting GCGM to genome-wide prediction is not straightforward and there is not a direct and principled method to cope with this problem. Therefore, some *ad hoc* assumptions have to be done in order to

provide a frequentist formulation. The method proposed here is named Graphical Maximum Likelihood-BLUP (GML-BLUP) because it involves two steps, combining the EM algorithm (Dempster et al., 1977) with GCGM to estimate covariance components (an algorithm denoted as G-EM) and plugging this estimates into mixed model equations corresponding to model 2.5 to obtain the empirical BLUP of $\boldsymbol{g}$ (Henderson, 1950; 1963). According to the theory of GCGM presented in subsection 2.1.1, the objective function 2.1 is strictly convex if the number of observations is greater than the size of the largest clique of G denoted as C. To meet this requirement under the rationale of the EM algorithm, the following approach is used. Suppose that data can be split into $f > C$ groups such that each group has a different vector of marker effects, that is, $\boldsymbol{y}_i = W_i\boldsymbol{g}_i + \boldsymbol{e}_i, \forall\, i = 1,2, \dots, f$. It is also assumed that: $\boldsymbol{g}_1, \dots, \boldsymbol{g}_f$ are iid $MVN(0, \Omega^{-1})$, $\boldsymbol{e}_1, \dots, \boldsymbol{e}_f$ are independent $MVN(0, \sigma^2 I_{n_i})$ random variables and $Cov(g_i, e_{i'}) = 0, \forall\, 1 \le i, i' \le f$, where n_i is the number of observations in group i; therefore, $\sum_{i=1}^{f} n_i = n$. A natural criterion to split data is to cluster individuals in half-sib or full-sib families as is commonly done in quantitative genetics (Falconer and Mackay, 1996). The assumption of heterogeneity of marker effects across families was also considered by Gianola et al. (2003). Under these assumptions, the complete log-likelihood can be written as:

$$l(\sigma^2, \Omega) = constants - \frac{n}{2}log\sigma^2 + \frac{f}{2}\left(log|\Omega| - tr(\Omega S_g)\right) - \frac{\|\boldsymbol{y} - W^*\boldsymbol{g}^*\|_2^2}{2\sigma^2} \qquad (2.7)$$

$$S_g = \frac{1}{f}\sum_{i=1}^{f} \boldsymbol{g}_i\boldsymbol{g}_i', \boldsymbol{g}^* := \left(\boldsymbol{g}_1' \cdots \boldsymbol{g}_f'\right)'$$

where W^* is a block diagonal matrix with blocks $W_1, \dots, W_f$.

Following Searle et al. (1992), the expected values of sufficient statistics for the covariance parameters taken with respect to the conditional distribution of the missing data given the observed data have to be found. From 2.6 it follows that the sufficient statistic for $(\Omega, \sigma^2) := \boldsymbol{\theta}$ is $(S_g, \boldsymbol{e}^{*\prime}\boldsymbol{e}^*), \boldsymbol{e}^* = \boldsymbol{y} - W^*\boldsymbol{g}^*$. It is easily shown that given $\boldsymbol{y}$, $\boldsymbol{g}_1, \dots, \boldsymbol{g}_f$ are independent with the

following distributions: $\boldsymbol{g}_i|\boldsymbol{y}_i \sim MVN\left(K_i^{-1}\frac{W_i'\boldsymbol{y}_i}{\sigma^2}, K_i^{-1}\right)$, where $K_i := \frac{W_i'W_i}{\sigma^2} + \Omega$. Similarly, it follows

that $\boldsymbol{e}^*|\boldsymbol{y} \sim MVN\left(\sigma^2 V^{-1}\boldsymbol{y}, \sigma^2(I - \sigma^2 V^{-1})\right)$, where $V = W^{*'}I_f \otimes \Omega^{-1}W^* + R$. Hence,

$$E[S_g|\boldsymbol{y}] = \frac{1}{f}\sum_{i=1}^{f} K_i^{-1}\left[I_m + \frac{1}{(\sigma^2)^2}W_i'\boldsymbol{y}_i\boldsymbol{y}_i'W_iK_i^{-1}\right] \tag{2.7}$$

$$E[\boldsymbol{e}^{*'}\boldsymbol{e}^*|\boldsymbol{y}] = \sigma^2(n - \sigma^2 tr(V^{-1}) + \sigma^2 \boldsymbol{y}'V^{-1}V^{-1}\boldsymbol{y}) \tag{2.8}$$

Applying the Woodbury's identity, $E[S_g|\boldsymbol{y}]$ can be alternatively expressed as:

$$E[S_g|\boldsymbol{y}] = \frac{1}{f}\Omega^{-1}\left\{fI_m - \left[\sum_{i=1}^{f} W_i'V_i^{-1}(I_{n_i} - \boldsymbol{y}_i\boldsymbol{y}_i'V_i^{-1})W_i\right]\Omega^{-1}\right\} \tag{2.9}$$

where $V_i := W_i\Omega^{-1}W_i' + \sigma^2 I_{n_i}$.

In 2.7, f $m \times m$ matrices have to be inverted and these matrices are dense because $W_i'W_i$ is

dense. On the other hand, in 2.9, Ω, which is sparse in GCGM, has to be inverted along with f $n_i \times n_i$

matrices. The expectation step of the G-EM algorithm consists of using either 2.7 or 2.9 to compute

$E[S_g|\boldsymbol{y}]$ and 2.8 to compute $E[\boldsymbol{e}^{*'}\boldsymbol{e}^*|\boldsymbol{y}]$, the maximization step is the one involving GCGM. At iteration

t, the maximization step involves the following computations:

$$(\hat{\sigma}^2)^{(t+1)} = \frac{\widehat{\boldsymbol{q}}^{(t)}}{n}, \widehat{\boldsymbol{q}}^{(t)} := E[\boldsymbol{e}^{*'}\boldsymbol{e}^*|\boldsymbol{y}]\Big|_{\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}}$$

$$\widehat{\Omega}^{(t+1)} = h\left(\hat{S}_g^{(t)}\right), \hat{S}_g^{(t)} := E[S_g|\boldsymbol{y}]\Big|_{\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}}$$

where function $h(\cdot)$ is as defined in subsection 2.1.1, it involves evaluating expressions 2.3 and 2.4

taking $\hat{S}_g^{(t)}$ as argument. Subsequently, matrices $\hat{L}^{(t+1)}$ and $\widehat{D}^{(t+1)}$ are built and $\widehat{\Omega}^{(t+1)}$ is computed as

$\hat{L}^{(t+1)}\widehat{D}^{(t+1)}\hat{L}'^{(t+1)}$. Once the algorithm converges and the maximum likelihood estimates of Ω and σ^2

are obtained, these are plugged in the mixed model equations corresponding to model 2.5 to obtain the

empirical BLUP of $\boldsymbol{g}$ (Henderson, 1950; 1963):

$$\widehat{\boldsymbol{g}} = \left(W'W + \hat{\sigma}^2\widehat{\Omega}\right)^{-1}W'\boldsymbol{y}.$$

2.4 Some criteria to define G and to order markers

A graph defining the partial correlation (the correlation between the effects of two markers given the effects of all the remaining markers, it can be computed from the elements of Ω) structure can be specified according to several genetic criteria. These criteria come from the assumptions about how marker allele substitution effects could be partially correlated. In the following, some alternatives to define G based on genetic and biological principles are discussed.

1. The first approach is based on the idea that there exists spatial correlation (Gianola et al. 2003, Yang and Tempelman 2012). In this case, the physical/linkage map of the considered loci is used to define G. One simple alternative is to define a window on the basis of a given physical/linkage distance or a given number of markers and slide it across each chromosome. The ordering of markers is the natural ordering imposed by the physical/linkage map. If markers are not equally distanced, for each chromosome it induces a differentially banded precision matrix in the first case and a banded matrix in the second case. Then the precision matrix for all marker effects is the direct sum of banded or differentially banded matrices (one per each chromosome).

2. A second approach to define the groups of loci whose effects are correlated is to use existing gene networks built using the same set of loci being considered. This would induce a graph G with a general structure, general in the sense that *a priori* it is not expected to follow a particular pattern. In this case, the ordering can be also dictated by spatial location of the loci.

3. A third approach is bringing biological information using bioinformatics tools to group the loci according to their function and/or the metabolic pathways in which they are known to be involved. This will create "functional groups" or "functional blocks" of loci, and there are two ways to define the graph.

3.a The allelic substitution effects of loci in the same group are partially correlated with each other. Correlation among effects of loci in different groups is not allowed. If the ordering

is such that groups are ordered first and then markers are ordered within group such that markers in groups with smaller index have smaller indexes, it induces a block-diagonal precision matrix.

3.b As in 3.a, but correlations among loci in different groups are allowed. The ordering is discussed later.

4. Lastly, some of the existing metrics to assess LD between loci can be used to define G by imposing a threshold and those pairs of loci having a metric greater than the threshold will be assumed to have partially correlated effects.

Because of the aforementioned properties, decomposable graphs have been object of research and whenever possible, it is desirable to work with this kind of graphs. Thus, some rules to define G that guarantee decomposability are presented as well. The following proposition establishes which approaches will induce decomposable graphs. Hereinafter, the "functional blocks" mentioned in approach 3 will be referred to as blocks. If a block contains a subset of markers with effects correlated with the effects of a subset of markers in another block, these blocks are said to be linked. Let B be the total number of blocks and $\mathcal{L}$ be the set of pairs of linked blocks. Let Ψ be the set of blocks linked with at least two other blocks, $\forall\, l \in \Psi$ let Γ_l be the set of blocks linked to block l and $\forall a \in \Gamma_l$, let C_{l_a} be the subset of markers in block l whose effects are correlated with effects of a subset of markers in block $a, 1 \leq a \leq B, a \neq l$.

Proposition 1 The graphs induced under approaches 1 and 3.a are decomposable and the graph induced under 3.b is decomposable if there exists an ordering of markers σ' that along with the edges set satisfy the following conditions.

Condition 1.1 For all possible triplets of linked blocks $\{l, l', l''\}$ such that $C_{l_{l'}} \neq C_{l_{l''}}$, $C_{l'_l} \neq C_{l'_{l''}}$, $C_{l''_l} \neq C_{l''_{l'}}$, and sets $I_l := C_{l_{l'}} \cap C_{l_{l''}}$, $I_{l'} := C_{l'_l} \cap C_{l'_{l''}}$ and $I_{l''} := C_{l''_l} \cap C_{l''_{l'}}$, are all non-empty, the following never happens: $\sigma'(i) > \sigma'(j) > \sigma'(k)$, $i \in C_{l_{l''}} \cap I_l^c, j \in C_{l'_l}$ or $i \in$

$C_{l_{l''}}$, $j \in C_{l'_l} \cap I_{l'}^C$, and $k \in I_{l''}$; if there are triplets of linked blocks $\{l, l', l''\}$ such that exactly one of the three sets $\{I_l, I_{l'}, I_{l''}\}$, say I_l is empty, then: $\min\{\sigma'(k), \sigma'(i), \sigma'(j)\} = \sigma'(k), \forall\, k \in C_{l_{l'}} \cup C_{l_{l''}} \forall\, j \in I_{l'} \forall\, i \in I_{l''}$ and if exactly two of these sets, say $\{I_l, I_{l'}\}$ are empty, then for either l or l', say l, $\sigma'(k) < \sigma'(i) \forall k \in C_{l_{l'}} \cup C_{l_{l''}} \forall i \in I_{l''}$. Superindex C indicates the complement with respect to the index set of the corresponding block.

Condition 1.2 For every possible triplet of blocks $\{l, l', l''\}$ the following does not happen: $\sigma'(k) < \sigma'(j) < \sigma'(i), k \in I_l, j \in C_{l'_l}, i \in C_{l''_l}, C_{l'_{l''}} = \emptyset$.

Condition 1.3 For every duplet of linked blocks $\{l, l'\}$ the following does not hold: $\exists\, i \in l, \{j, k\} \in l'$ such that $\sigma'(i) > \sigma'(j) > \sigma'(k), i \in C_{l_{l'}}, j \in C_{l_l}^C, k \in C_{l'_l}$.

Condition 1.4 For each pair of linked blocks (l, l'), $C_{l_{l'}} \times C_{l'_l} \in E_\sigma$, that is, the effect of each marker in $C_{l_{l'}}$ is correlated with the effects of all marker in $C_{l'_l}$.

Moreover, conditions 1.1, 1.2 and 1.3 are necessary whereas condition 1.4 is not.

Proof See appendix B.

This proposition involves all possible orderings of markers. However, if markers are ordered in such a way that markers in the same block are given consecutive indices, the number of possible orderings is reduced. Thus, in order to provide a simpler way to order markers, the following proposition only requires the existence of an ordering of the blocks and a structure on the edges set satisfying certain conditions that permit to find a perfect elimination ordering of markers.

Proposition 2 If there exists an ordering ρ of the blocks which coupled with the structure of the edges set satisfy condition 1.4 plus the following conditions:

Condition 2.1 $C_{l_a} = \cdots = C_{l_m} := C_l \forall l \in \Psi$

Condition 2.2 For every possible triplet of blocks $\{l, l', l''\}$ the following does not happen: $(l, l'), (l, l'') \in L, (l', l'') \notin L, \rho(l) < \rho(l') < \rho(l'')$.

Then the following ordering strategy (denoted by σ) of marker loci is a perfect elimination ordering: once blocks have been ordered according to ρ, markers are ordered in such a way that the smaller the index of a block the smaller the indices of the markers pertaining to that block. The ordering inside each block is done as follows: markers in C_l are given the largest indices in block l. In addition, under this ordering strategy, condition 2.2 is also necessary for σ to be a perfect elimination ordering whereas condition 2.1 is not.

Proof See Appendix C.

Corollary to Proposition 2 Consider the "super graph" formed by regarding the blocks as super nodes and $\mathcal{L}$ as a "super vertices set". Then, under conditions 2.1 and 1.4, if the "super vertices set" admits a perfect elimination ordering, the ordering defined in proposition 2 corresponds to a perfect elimination scheme.

Proof See Appendix D.

2.5 Simulated and real data analyses

To ensure computational tractability, a small genome comprised of a single chromosome of 1 M length, with 1000 biallelic QTL and 3201 SNP markers was simulated through a forward-in-time approach using the software QMSim (Sargolzaei and Schenkel, 2013). A historical population was simulated by creating 1000 generations of random mating in order to reach mutation-drift equilibrium and to create LD (Sargolzaei and Schenkel, 2013). Each generation of the historical population consisted of 500 males and 500 females. Markers whose minor allele frequency (MAF) in the last historical generation was smaller than 0.05, were discarded. Then, frequencies of selected markers in each generation were checked in order to remove those with fixed alleles. Across generations, all markers had MAF greater than 0.01. Phenotypes were simulated as the sum of additive genetic values and an error term. Errors were drawn from independent normal distributions with null mean and variance equal to 50; three heritability values were considered: 0.01, 0.5 and 0.86. Two strategies to

21

simulate the QTL allelic effects were considered. The first one involved sampling the effects from independent $N(0,1)$ distributions whereas in the second one QTL effects were drawn from a $MVN(0, M)$ distribution, where M^{-1} was a banded diagonal matrix with band size 10. In both cases, QTL effects were scaled to obtain the appropriate additive genetic variances. The same QTL and SNP genotypes were used in both strategies. Simulation of errors, and QTL effects for strategy 2 as well as all the statistical analyses were performed using existing R packages (Matrix, mvtnorm, pscl) and in-house R scripts (R Core Team, 2015). Ten replicates of each simulation were obtained and only decomposable graphs were considered. The graph G was built using approach 1, considering both, windows defined by a fixed number of marker loci (6) or a fixed map distance (0.1875 cM). When windows were based on number of markers models were denoted as Bayes G-D_B and GML-BLUP_B and when windows were defined by a given map distance models were denoted as Bayes G-D_DB and GML-BLUP_B. These window sizes were defined on the basis of the average LD decay (using r^2 as a measure of LD), considering the distance at which the average LD was smaller than 0.1. Also, some conventional models used in genome-wide prediction were fitted; Bayes A and Bayesian Ridge Regression (BRR). Three sets of analyses were performed. The first one corresponded to fitting models considering SNP effects to datasets simulated with strategy 1 (scenario 1). In the second one, the same models were fitted to datasets from simulation strategy 2 (scenario 2). Finally, the third set involved fitting our models to datasets from strategy 2, but models considered QTL instead of SNP effects (scenario 3, the ideal scenario). In each replicate, the training set was defined as generations 0, 1 and 2, and generation 3 was the validation set. Therefore, in these simulations we performed forward prediction (one generation ahead).

Real data analyses were performed for three different traits using a small dataset from the University of Florida's Beef Research Unit (BRU). Briefly, the three phenotypes were daily feed intake (DFI), the ultrasonic measure of percent of intra-muscular fat (UPIMF) and live weight taken the same

day than UPIMG (UW). These records came from individuals genotyped with the Illumina GoldenGate Bovine3K BeadChip (Illumina, Inc., 2011) belonging to a multibreed Angus-Brahman herd. For details on the multibreed herd, the genotypes and the traits see Elzo et al. (2012) and Elzo et al. (2013). The heritabilities of these traits estimated in the multibreed herd were 0.31 for DFI (Elzo et al., 2012), 0.54 for UW and 0.53 for UPIMF (Elzo et al., 2013). The same models fitted to simulated data except those with differentially banded structure were fitted to these traits after correcting data for the following fixed effects: contemporary group (year-pen), age of dam, age of individual, sex, Brahman fraction and heterozygosity of the individual. After editing data such that only individuals with complete genotypes were included, the final dataset had 102 individuals and 2407 SNP. The training set included the oldest 72 individuals, and the validation set was composed of the 30 remaining individuals. In this case, the windows used to build G (which determine the bandwidth of the precision matrix) were defined such that they contained 5 adjacent markers or based on observed LD. In the latter case, for each chromosome, the number of SNP defining the bandwidth of the precision matrix was chosen using the average distance for which the LD was larger than 0.1. Genotypes for denser chips are available in this herd, but this small dataset was used to ensure computational tractability. Only within chromosome correlations were allowed.

In all frequentist analyses, individuals were grouped according to half-sib families. Models were compared through the Pearson correlation of phenotypes and predicted additive genetic values in the validation set (hereinafter predictive accuracy) for real and simulated data, and the Pearson correlation between true and predicted additive genetic values (accuracy of predicted additive genetic values) in training and validation sets for simulated data. For Bayesian models, a total of 25000 MCMC samples were obtained using the Gibbs sampler described in subsection 2.2 considering the first 10000 samples as burn in. For the GML-BLUP models, convergence was declared when the change in the average absolute difference of the parameters in two consecutive iterations was smaller than 1×10^{-4}.

3. RESULTS AND DISCUSSION

In this study, Bayesian and frequentist methods to consider partially correlated marker effects in genome-wide prediction using GCGM were developed. Several approaches, based on genetic principles, to define the graph G were proposed and it was shown that some of these approaches induced a decomposable graph (some of them requiring particular conditions). The performance of the proposed models was addressed in small simulated and real datasets. Two of the conventional hierarchical Bayesian models used in genome-wide prediction (Bayes A and BRR) were also fitted to these data. Simulated data consisted of six sets (the combination of two strategies to simulate QTL effects and three heritability values) of ten replicates each. These sets shared the simulated QTL and SNP genotypes. In three of these sets, QTL effects were sampled from independent Gaussian distributions, while in the remaining sets these were sampled from a multivariate Gaussian distribution with a banded precision matrix. Analyses were performed under three scenarios using these sets of simulated data. In scenario 1, it was expected that correlations arose from physical proximity to causal variants. On the other hand, in scenario 2, the fact that QTL effects were correlated introduced another potential source of correlation among SNP effects. Lastly, in scenario 3, phenotypes were regressed on QTL genotypes instead of SNP genotypes; thus, under this ideal scenario it was expected *a priori* that our models would outperform standard models assuming independence. It is said that this is an ideal scenario because the causal variants are considered in the model instead of proxies like molecular markers.

3.1 Predictive accuracies and accuracies of predicted additive genetic values in simulated and real datasets

Table 1 shows the summary of the performance of the different models considered here in simulated datasets under scenarios 1, 2 and 3 when heritability was 0.5.

24

Table 1 Average (over 10 replicates) predictive accuracies (APA) and accuracies of predicted additive genetic values in training (AAT) and validation (AAV) sets for simulated datasets when heritability was 0.5. Standard deviations are presented inside the brackets.

Model	Scenario 1			Scenario 2			Scenario 3		
	APA	AAT	AAV	APA	AAT	AAV	APA	AAT	AAV
Bayes G-D_B	0.428	0.750	0.588	0.386	0.740	0.556	0.512	0.791	0.710
	(0.058)	(0.032)	(0.090)	(0.080)	(0.025)	(0.081)	(0.055)	(0.025)	(0.043)
Bayes G-D_DB	0.424	0.750	0.587	0.380	0.737	0.537	--	--	--
	(0.061)	(0.028)	(0.086)	(0.083)	(0.023)	(0.081)			
GML-BLUP_B	0.540	0.805	0.730	0.536	0.800	0.728	0.577	0.830	0.771
	(0.065)	(0.033)	(0.068)	(0.054)	(0.027)	(0.037)	(0.038)	(0.025)	(0.028)
GML-BLUP_DB	0.530	0.804	0.722	0.547	0.810	0.744	--	--	--
	(0.062)	(0.030)	(0.071)	(0.050)	(0.026)	(0.036)			
Bayes A	0.341	0.736	0.463	0.304	0.720	0.440	0.478	0.735	0.630
	(0.054)	(0.025)	(0.081)	(0.101)	(0.024)	(0.092)	(0.038)	(0.017)	(0.071)
BRR	0.453	0.764	0.616	0.439	0.739	0.619	0.504	0.780	0.706
	(0.062)	(0.027)	(0.096)	(0.069)	(0.024)	(0.071)	(0.046)	(0.026)	(0.044)

In scenario 1, APA, AAT and AAV were higher for GML-BLUP models, whereas Bayes A had the poorest performance; on the other hand, BRR and Bayes G-D performed similarly. Likewise, in scenario 2, the GML-BLUP models had the best predictive performance and Bayes A had the worst; however, this time the difference between Bayes G-D and BRR was more marked, the latter model outperformed Bayesian models accounting for correlated effects. Finally, in scenario 3, the performance of all models was more similar than in the previous cases. Again, GML-BLUP had the highest APA, AAT and AAV and Bayes A had the lowest, but this time Bayes G-D slightly outperformed BRR. In general, Bayes A showed the worst performance and GML-BLUP the best one, and there were tight differences between models considering banded precision matrices and those considering differentially banded precision matrices. Despite of the fact that whole-genome regression models like those considered here can capture genetic relationships among genotyped individuals (Habier et al., 2007), a factor that could be behind the superiority of the frequentist models is the explicit use of family information when estimating Ω (as explained in section 2.3). On the other hand,

Bayes A assumes independent marker effects, each one having a different variance, and for these datasets this assumption yielded the worst results. When moving from scenario 1 to scenario 2, there was not an increment in the overall performance of models accounting for partial correlation, in fact, for Bayesian models there was a light drop in APA, AAT and AAV. A potential explanation is the fact that the bandwidths of the precision matrix were defined on the basis of LD between pairs of markers and were not tuned. Tuning was not performed due to limited computational resources. Recall that the concentration matrix of marker effects was unknown, because QTL instead of SNP effects were simulated. Consequently, misspecification of the partial correlation structure may have been another possible cause of this result, i.e., perhaps the partial correlation matrix of SNP effects was not banded. When simulated QTL effects were partially correlated and models were parameterized in terms of these effects (scenario 3), all models accounting for partially correlated effects performed better than Bayes A and BRR; however, in terms of APA, differences between Bayes G-D and BRR were small. As expected, the performance of all models improved markedly when fitting QTL instead of marker effects, the increment being more visible for Bayes G-D. In these simulations, the effect of factors such as number of QTL controlling the trait and number of SNP it were not explored.

For the low heritability trait, Bayes G-D_B had the highest APA, AAT and AVV in the three scenarios and GML-BLUP_B did not show the superiority exhibited for the high heritability trait. Table 2 summarizes the results of this analysis.

Table 2 Average (over 10 replicates) predictive accuracies (APA) and accuracies of predicted additive genetic values in training (AAT) and validation (AAV) sets for simulated datasets when heritability was 0.01. Standard deviations are presented inside the brackets.

Model	Scenario 1			Scenario 2			Scenario 3		
	APA	AAT	AAV	APA	AAT	AAV	APA	AAT	AAV
Bayes G-D_B	-0.008 (0.097)	0.208 (0.082)	0.163 (0.061)	0.029 (0.082)	0.183 (0.072)	0.157 (0.096)	0.033 (0.077)	0.227 (0.071)	0.155 (0.048)
Bayes G-D_DB	-0.007 (0.098)	0.211 (0.071)	0.162 (0.060)	0.030 (0.080)	0.185 (0.073)	0.159 (0.096)	--	--	--

Model	APA	AAT	AAV	APA	AAT	AAV	APA	AAT	AAV
GML-BLUP_B	-0.011	0.192	0.132	0.001	0.171	0.120	-0.046	0.166	0.119
	(0.085)	(0.069)	(0.104)	(0.083)	(0.067)	(0.091)	(0.090)	(0.083)	(0.079)
GML-BLUP_DB	-0.011	0.193	0.130	0.001	0.170	0.121	--	--	--
	(0.084)	(0.067)	(0.110)	(0.082)	(0.066)	(0.089)			
Bayes A	-0.011	0.187	0.144	-0.005	0.165	0.123	0.001	0.198	0.122
	(0.092)	(0.116)	(0.082)	(0.090)	(0.118)	(0.089)	(0.114)	(0.078)	(0.078)
BRR	-0.015	0.187	0.156	-0.004	0.167	0.125	-0.011	0.207	0.139
	(0.089)	(0.100)	(0.076)	(0.088)	(0.101)	(0.079)	(0.092)	(0.087)	(0.068)

Due to the low heritability of this trait, there is little benefit from the explicit use of family information in GML-BLUP and this is what was observed in this analysis. Results for the high heritability trait are presented in Table 3.

Table 3 Average (over 10 replicates) predictive accuracies (APA) and accuracies of predicted additive genetic values in training (AAT) and validation (AAV) sets for simulated datasets when heritability was 0.86. Standard deviations are presented inside the brackets.

Model	Scenario 1			Scenario 2			Scenario 3		
	APA	AAT	AAV	APA	AAT	AAV	APA	AAT	AAV
Bayes G-D_B	0.895	0.956	0.932	0.896	0.955	0.930	0.911	0.954	0.941
	(0.069)	(0.042)	(0.024)	(0.068)	(0.045)	(0.026)	(0.071)	(0.040)	(0.025)
Bayes G-D_DB	0.890	0.957	0.932	0.887	0.953	0.931	--	--	--
	(0.071)	(0.044)	(0.022)	(0.070)	(0.043)	(0026)			
GML-BLUP_B	0.916	0.975	0.957	0.924	0.977	0.959	0.926	0.971	0.959
	(0.068)	(0.050)	(0.038)	(0.070)	(0.049)	(0.039)	(0.070)	(0.069)	(0.037)
GML-BLUP_DB	0.917	0.977	0.956	0.925	0.0976	0.956	--	--	--
	(0.067)	(0.052)	(0.036)	(0.069)	(0.050)	(0.040)			
Bayes A	0.880	0.939	0.890	0.881	0.930	0.889	0.906	0.950	0.933
	(0.073)	(0.075)	(0.074)	(0.070)	(0.075)	(0.074)	(0.070)	(0.068)	(0.072)
BRR	0.900	0.957	0.911	0.901	0.956	0.901	0.909	0.951	0.937
	(0.072)	(0.080)	(0.073)	(0.073)	(0.076)	(0.075)	(0.071)	(0.073)	(0.074)

In this case, results followed were similar to those observed for the medium heritability trait. GML-BLUP had the best performance, followed by models BRR and Bayes G-D. Moreover, model Bayes A had the worst performance. In general, when the simulated trait had a medium or high heritability (Tables 1 and 3), model GML-BLUP was superior in all scenarios; however, when heritability was

high, differences between predictive performances of models were less marked. Furthermore, when heritability was low, our frequentist approach GML-BLUP did not yield the best performance. This may be evidence supporting the idea that for phenotypes with medium to high heritabilities, the fact that GML-BLUP uses family information explicitly may result in a higher predictive capability. Although we only considered three heritability values, these results suggest that the relative performance of this set of models may change with the heritability of the trait and therefore, it is an issue that has to be studied in future research. Table 4 contains the predictive accuracies for the three traits in the real dataset.

Table 4 Predictive accuracies for real data analyses

Model[1]	DFI	UPIMF	UW
Bayes G-D_B	0.20	-0.17	0.03
Bayes G-D_B*	0.10	0.04	-0.04
GML-BLUP_B	0.02	0.09	-0.06
GML-BLUP_B*	-0.02	0.03	-0.04
Bayes A	0.08	-0.06	-0.007
BRR	-0.18	-0.19	-0.17

[1]The asterisk indicates that the bandwidth of the precision matrix was defined on the basis of LD.

In this real dataset, for DFI, Bayes G-D assuming a banded matrix with a bandwidth of 5 yielded the highest predictive accuracy and BRR the lowest. Bayes G-D assuming a bandwidth defined on the basis of LD had the second highest predictive accuracy. For UPIMF, GML-BLUP_B had the best performance, followed by Bayes G-D_B* and GML-BLUP_B*, model BRR had the lowest. Finally, for UW, the trait with the lowest predictive accuracies, Bayes G-D_B outperformed the other models which had similar predictive accuracies, all negative and close to zero. For DFI, predictive accuracies from the Bayesian models proposed in this study were higher than those obtained by Elzo et al. (2012) for the same population, but using a larger sample size, a higher number of markers and models that considered: only marker effects, marker and polygenic effects, and only polygenic effects. Albeit the estimated heritabilities of these traits were from moderate to high, predictive accuracies were low. It is known that the predictive performance of genome-wide selection is affected by sample size

and number of markers (Goddard and Hayes, 2002; Goddard and Hayes, 2007; Goddard, 2009; Zhong et al., 2009). Consequently, the low predictive abilities observed here could be the result of the low marker density and the small sample size. Although these results need to be validated in a larger population and using a higher number of markers, they suggest a superiority of models developed here over models assuming independent marker allele substitution effects as well as a marked difference in the performance of models assuming different bandwidths of the precision matrices. Model BRR always yielded negative predictive accuracies.

Yang and Tempelman (2012) found increments in accuracy of additive genetic values in the testing population up to 3% in simulated datasets and 3.6% in a real dataset when comparing their first order antedependence models with their independent marker effects counterparts Bayes A and Bayes B. The simulations carried out in Yang and Tempelman (2012) are closer to those considered in scenario 1, where it is expected that correlation among SNP arises from physical proximity to the causal variants. The heritability used in their simulations was also 0.5 and they also simulated a genome formed by a single chromosome of 1 M length. In this study, differences in accuracy in the testing population in scenario 1 were larger than those found by Yang and Tempelman (2012). The ratio of number of SNP to number of QTL was 3.2, perhaps it was too low for SNP effects to be correlated due to linkage with the same causal variants. In Yang and Tempelman (2012) these ratios ranged from 3.2 to 72.73, and they considered only 30 QTL. Yang and Tempelman (2012) reported improvements in accuracy in simulated and real datasets when comparing the antedependence models with their counterparts Bayes A and Bayes B; however, their results showed that Bayes B outperformed ante-Bayes A in terms of accuracy for simulated data and predictive accuracy for the mice data set described in Legarra et al. (2008). The authors attributed the observed results for simulated data to the small number of simulated QTL. The study of Gianola et al. (2003) did not consider data analyses.

3.2 Some comments about the models

Our models account for partially correlated SNP effects through imposing the Markov property with respect to G over these random variables. The graph is defined based on available prior information on the genetic architecture of the trait. This specification might be crucial for the predictive performance of these models and it is an aspect that we did not explore in this paper that poses a very interesting topic for further research. Notice that the conventional genome-wide regression models assuming a diagonal covariance matrix, and therefore a diagonal precision matrix as well; are also imposing a conditional covariance structure. This corresponds to the especial case when $E = \emptyset$, that is, the edges set is the empty set. An extension to be considered in future studies is to use the richer family of priors developed by Letac and Massam (2007) which is known as the type II Wishart family. It is more flexible as it permits to include a $k + 1$-dimensional shape parameter where k is the number of cliques of G. It allows inducing differential shrinkage trough the precision matrix. Recall that the priors used in this study have a single shape or shrinkage parameter δ. The only limitation of the type II Wishart family of priors is that it is restricted to decomposable graphs; nevertheless, recent research by Ben-David et al. (2015) has extended this flexible family of priors for the case of general DAG's and their family is even more flexible because it has as many shape parameters as the DAG has nodes.

As noticed in subsection 2.3, formulating a frequentist solution to the problem considered in this study is not as straightforward as doing it from the Bayesian perspective. Hence, this is another example of a problem that can be easily solved using the Bayesian paradigm, but whose frequentist solution is more complicated. Taking advantage of hierarchical models, which allow modeling complex processes (Lehmann and Casella, 1998), it was possible to provide a principled and simple solution from the Bayesian side. On the other hand, in order to find a frequentist solution, it was necessary to develop an *ad hoc* procedure that involves a partition of data which permits the computation of the MLE of the precision matrix. It is based on the assumption that different families have different sets of marker effects. It is necessary because otherwise function 2.1, which is used in

the M step of the G-EM algorithm, is not strictly convex and therefore a global minimum does not exist. Under the assumption that $\boldsymbol{g}$ is a random, the problem is to estimate the realization of this random vector that determined the observed data (Henderson, 1984). If there is a single vector $\boldsymbol{g}$, it is assumed that this realization is the same for the whole population. However, if marker allele substitution effects are random, it could be possible that the realization varies across certain groups or even across individuals. When formulating a mixed effects model to account for correlated marker effects, Gianola et al. (2003) also considered the possibility of having different marker effects across families. In the GML-BLUP approach, this idea is only used to estimate Ω because later on, this MLE is plugged in the mixed model equations corresponding to model 2.5 which assumes a single $\boldsymbol{g}$. From a pragmatic point of view, the assumptions of heterogeneous marker effects across families can be seen as a trick and thinking of this GML-BLUP procedure as a predictive machine, it is worth to assess its predictive performance, which indeed was satisfactory in the small datasets considered here. Of course, empirical BLUP for family-specific vectors $\boldsymbol{g}_1, \dots, \boldsymbol{g}_f$ under the assumptions presented in subsection 2.3 can be easily obtained using the appropriate mixed models equations and it could be addressed if the accuracy and predictive accuracy of this alternative approach are better.

When using mixed model methodologies to compute the BLUP of marker effects, the inverse covariance matrix has to be found in order to build the mixed model equations. In the classical animal model, the so-called Henderson rules (Henderson, 1976) allowed an efficient computation of this matrix. In the case of a model parameterized in terms of markers as in VanRaden (2008), finding the inverse covariance matrix is not an issue because it is diagonal. When allowing partially correlated effects, this matrix is no longer diagonal, and although it is sparse, there are no rules to avoid direct inversion; however, an advantage of the GEM-BLUP method is that the inverse covariance matrix is obtained directly, so there is no need for inversion. This feature can be seen as another advantage of using GCGM in genome-wide prediction.

3.3 Building graph G

The criteria to define G presented in subsection 2.4 may not reflect the true underlying conditional covariance structure; notwithstanding, it could capture some partial correlations that might improve the accuracy of genome-wide prediction. Of course, an ideal scenario would be to identify the real underlying conditional covariance structure which would be expected to improve the predictive performance of the models. At this point, it is worth to mention methods that "let the data talk" to define G. These methods exist in the theory of GCGM and as mentioned in the introduction, this is a model selection problem where the sparsity pattern of the precision matrix has to be found and the non-zero entries have to be estimated (Bickel and Levina 2008; Rajaratnam et al., 2008; Khare et al., 2015). In ongoing research, we are also developing methods for model selection that automatically find the graph G using frequentist and Bayesian approaches.

Here, some strategies to define the graph G bringing biological information were proposed; however, these were not implemented in the real data analyses because they imply a formidable challenge at the bioinformatics and biochemical level which is beyond the scope of this study. Therefore, evaluating the performance of our models when G is defined through the use of biological information is another topic for further research. The use of biological information proposed here involves the use of bioinformatics tools to group markers based on features such as their annotated function, i.e., using gene annotation information.

The methods presented in Gianola et al. (2003) assumed a spatial correlation depending on distance between markers, considered only within chromosome correlations and some of them required equally spaced markers. On the other hand, the methods of Yang and Tempelman (2012) assumed a non-stationary correlation structure based on the spatial position of the markers (using physical or linkage maps). The methods proposed here are more versatile and flexible because they allow defining the partial correlation structure based on criteria different to physical position. In fact, our methods are a

way to incorporate biological information (e.g., knowledge of biochemical pathways.) into statistical models for genome-wide prediction through the use of such information to define the conditional covariance structure of marker allele substitution effects. Moreover, our models can take into account the fact that partial correlation of loci effects in different chromosomes may exist. When using biological information to define G, markers are clustered in "functional sets" that were referred to as blocks. This can be done as described in Peñagaricano et al. (2013). Briefly, in a first step they assigned markers to genes and then they used existing databases (e.g., the Gene Ontology database) to define functional classes of genes. Thus, markers in the same functional class conform a block. For further details, see Peñagaricano et al., (2013). One potential limitation of this strategy is that annotation for all markers in a given chip may not be available; consequently, not all markers can be grouped. In this case, an easy solution is to assign these markers without information to a functional group based on physical distance as in Peñagaricano et al. (2013) or some LD metric. Alternatively, the effect of each marker in this set could be considered to be conditionally independent of the effects of the remaining markers. When allowing effects of markers in different blocks to be correlated, conditions for the induced graph to be decomposable are stated in proposition 1. The importance of this proposition is that these conditions can be seen as way to construct a decomposable graph under approach 3.b. This proposition requires the existence of an ordering that coupled with the edges set satisfy conditions 1.1 to 1.4. An ideal scenario would be the one where there exists an ordering of the markers that satisfies these conditions without modifying the edges set. However, if it does not exist, notice that it might be possible to satisfy these conditions by adding and deleting edges. Of course, it is not optimal because it implies removing important edges (i.e., partial covariances) and adding others that may not exist in the true underlying graph. Because it is not desirable to remove edges established on the basis of biological information, it might be better to add edges only. This would increase the number of covariance parameters in the model, but this is the price paid for finding a decomposable graph in order

to take advantage of the theoretical and computational advantages of this sort of graphs. Moreover, this can be seen as a compromise between the generality of allowing arbitrary partial correlations between effects of markers in different blocks and the case of conditionally independent blocks (i.e., no conditional covariances among effects of markers in different blocks). Notice that in the independent blocks case, the induced graph is also decomposable, but allowing correlations among marker effects in different blocks while maintaining the advantages of decomposability allows more generality in the modeling process and is more appealing from the biological perspective. Because of these reasons, we believe that this proposition could be useful.

For m variables, there are $m!$ possible orderings which may make the search for an ordering of markers satisfying conditions 1.1 to 1.3 intractable. Therefore, proposition 2 was developed in order to reduce the problem. Using this strategy, a perfect elimination ordering can be found just by manipulating blocks instead of individual markers. Thus, the task consists in finding an ordering of blocks satisfying condition 2.2 (provided the edges set satisfies 2.1 and 1.4). If such an order does not exist, then extra links between blocks could be added to satisfy this condition. However, this strategy is more restrictive in terms of the conditional covariance structure, as imposed by condition 2.1. This is the cost of having an easier way to find a perfect elimination ordering (whose existence implies decomposability). Under corollary to proposition 2, there are scenarios where it is very easy to order markers according to a perfect elimination scheme. For example, when each block is linked at most with one block, or when all blocks are linked, the order of blocks is arbitrary.

The idea of adding edges to a given graph G in order to find a decomposable graph containing it has been used in the realm of graphical models for covariance estimation; this decomposable super-graph is known as the cover of G (Khare and Rajaratnam, 2012). However, the problem of finding this graph is often computationally intractable and there are no general rules on how to proceed. Propositions 1 and 2, and the corollary to proposition 2 provide conditions ensuring decomposability

for the particular case described in approach 3.b and also provide rules to build decomposable graphs. In cases where assessing all possible orderings is not possible or there is no ordering that satisfies these conditions, they may guide the process of adding edges, i.e., may indicate what edges have to be added. Moreover, notice that conditions in proposition 2 are easier to assess than those in proposition 1 and that corollary to proposition 2 provides even simpler ways to find a perfect elimination ordering. Once more, notice that the price paid for easing the ordering of markers is imposing a more restrictive covariance structure. However, imposing highly restrictive structures has resulted in some improvements in accuracy. In Yang and Tempelman (2012), it was assumed that the effect of each marker was correlated only with the effects of adjacent markers according to the physical position, which is clearly a very restrictive assumption, yet they found some gains in accuracy as discussed before.

3.4 Multiallelic loci

Currently, there exists a widespread use of SNP chips to bring genomic information into the statistical models used to carry out prediction of genetic values and phenotypes in animals, humans and plants. These markers are used just as a proxy for the actual genes affecting the trait or set of traits under consideration and they are typically biallelic. However, advances in bioinformatics are allowing the determination of genotypes for actual genes. If all genes of a given species were known, predictions could be based on genotypes of these genes instead of marker genotypes, it has an obvious biological advantage because it is better to work with actual genes in lieu of proxies. Hopefully, such an ideal scenario is going to be attained at some point in the future. In this case, for many loci there would be more than two alleles. Another case in which multiallelic loci have to be considered is the following. Genome-wide prediction models have also considered diplotypes defined from haplotypes that have been built from two or more adjacent markers as explanatory variables (Meuwiseen et al., 2001; Calus et al., 2008); consequently, the number of haplotypes is greater than two. Of course, if the explanatory variables are built based on molecular markers, haplotypes, or actual genes, there are biological

implications. Notwithstanding, from a merely statistical point of view, the methods proposed here can be easily extended to the multi-allelic case. If there are a_h alleles at locus h, then the corresponding columns of the design matrix are built by defining $a_h - 1$ variables as follows:

$$W^h = \{w_{ij}^h\} = \begin{cases} 1, if\ genotype = A_j A_j \\ 0, if\ genotype = A_j -\ , j = 1,2,\dots,a_h - 1\ , \\ -1, if\ genotype = --\ \end{cases}$$

where w_{ij}^h is the "centered" gene content (i.e., gene content-1) of the ith individual for the jth allele and $"-"$ represents an allele different from A_j. Across loci, the graph G can be defined using approaches presented in subsection 2.4, but now it has to be considered that there are two or more effects per locus; consequently, the conditional covariance structure among effects of alleles at the same locus has to be defined. Perhaps, a simple approach is to assume that either, all effects at the same locus are partially correlated or that they have a null partial correlation.

4. CONCLUSIONS

An important contribution of this paper is the introduction of GCGM in the context of genome-wide prediction to account for partially correlated marker effects, a phenomenon that was known to happen since early stages of this technology and that has not been widely studied. Methods to incorporate this rich and useful theory in the problem of predicting genetic values using genomic data via linear models were developed. Our methods are flexible in the sense that they offer frequentist and Bayesian approaches, allow incorporating biological information in the prediction problem, and permit to model a wide range of conditional covariance structures resulting from different factors (not only markers' position in the genome). One possible way to make our methods more effective is through the use bioinformatics, biochemistry and physiology to build more reliable graphs which may significantly increase the accuracy of genome-wide prediction.

Author contributions

C.A. Martínez developed the methods, proved the propositions and the corollary, wrote most of the R scripts, designed and carried out the simulations and wrote the paper. K. Khare advised the development of methods, reviewed and discussed the proofs and statistical aspects of the paper. S. Rahman discussed the methods and wrote some of the R scripts. M.A. Elzo helped in designing the simulation, reviewed and discussed the genetic aspects of the paper.

Acknowledgements

<div align="center">

REFERENCES

</div>

Aguilar, I., Misztal, I., Johnson, D.L., Legarra, A., Tsuruta, S., & Lawlor, T.J. (2010). Hot topic: A unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. *Journal of Dairy Science, 93,* 743-752.

Ben-David, E., Li, T., Massam, H., & Rajaratnam, B. (2015). High dimensional Bayesian inference for Gaussian directed acyclic graph models. *arXiv preprint,* arXiv: *1109.4371 v5*.

Bickel, P.J., & Levina, E. (2008). Covariance regularization by thresholding. *The Annals of Statistics, 36(6)*, 2577-2604.

Calus M.P.L., Meuwissen, T.H.E., de Roos, A.P.W., & Veerkamp, R.F. (2008). Accuracy of genomic selection using different methods to define haplotypes. *Genetics, 178,* 553-561.

Carvalho, C.M., Massam, H., & West, M. (2007). Simulation of hyper-inverse- Wishart distributions in graphical models. *Biometrika, 94*, 647-659.

Chang, H.H., & McGeachie, M. (2011). Phenotype prediction by integrative network analysis of SNP and gene expression microarrays. *Conference Proceedings IEEE Engineering in Medicine and Biology Society 2011*, 6849–6852.

Dawid, A.P., & Lautitzen, S.L. (1993).Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics, 21(3)*, 1272-1317.

Dempster, A. (1972). Covariance selection. *Biometrics 28*, 157-175.

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

Diaconis, P., & Ylvisaker, D. (1979). Conjugate priors for exponential families. *The Annals of Statistics, 7(2)*, 269-281.

Elzo, M. A., Lamb, G.C., Johnson, D.D., Thomas, M.G., Misztal, I., Rae,D.O., Martinez, C.A., Wasdin, J.G., & Driver, J.D. (2012). Genomic-polygenic evaluation of Angus-Brahman multibreed cattle for feed efficiency and postweaning growth using the Illumina3k chip. *Journal of Animal Science, 90*, 2488-2497.

Elzo, M. A., Martinez, C.A., Lamb, G.C., Johnson, D.D., Thomas, M.G., Misztal, I., Rae,D.O., Wasdin, J.G., & Driver, J.D.. (2013). Genomic-polygenic evaluation for ultrasound and weight traits in Angus-Brahman multibreed cattle with the Illumina3k chip. *Livestock Science, 153*, 39-49.

Falconer, D.S., & Mckay, T.F.C. (1996). *Introduction to quantitative genetics* (4th ed.). Harlow, UK: Longmans Green.

Gianola, D., Perez-Enciso, M., & Toro, M.A. (2003). On marker-assisted prediction of genetic value: Beyond the Ridge. *Genetics, 163*, 347-365.

Gianola, D., de los Campos, G., Hill, W.G., Manfredi, E., & Fernando, R.L. (2009). Additive genetic variability and the bayesian alphabet. *Genetics,183*, 347-363.

Gianola, D. (2013). Priors in whole-genome regression: The Bayesian alphabet returns. *Genetics, 194*, 573-596.

Goddard, M.E., Hayes, B.J. (2002). Optimisation of response using molecular data. *Proceedings of the 7th World Congress on Genetics Applied to Livestock Production, 33*, 3–10

Goddard, M.E., Hayes, B.J. (2007). Genomic Selection. *Journal of Animal Breeding and Genetics*, 124, 323-330.

Goddard, M.E. (2009). Genomic selection: prediction of accuracy and maximization of long term response. *Genetica*, 136, 245-257.

Habier, D., Fernando, R.L., & Dekkers, J.C.M. (2007).The impact of genetic relationship information on genome-assisted breeding values. *Genetics, 177*, 2389-2397.

Han, B., Chen, X., Talebizadeh, Z., & Xu, H. (2012). Genetic studies of complex human diseases: Characterizing SNP-disease associations using Bayesian networks. *BMC Systems Biology, 6(3)*, S14.

Henderson, C.R. (1950). Estimation of genetic parameters. *The Annals of Mathematical Statistics, 21*, 309.

Henderson, C.R. (1963). Selection index and expected genetic advance. In W.D. Hanson & H.F. Robinson (Eds.), *National Academy of Sciences-National Research Council,* publication 982.

Henderson, C.R. (1976). A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. *Biometrics, 32*, 69-83.

Henderson, C.R. (1984). *Applications of linear models in animal breeding*. Guelph, ON: University of Guelph.

Illumina, Inc., 2011. GoldenGate Bovine3K Genotyping BeadChip. Illumina Data Sheet. San Diego, CA. www.illumina.com/Documents//products/datasheets/datasheet_bovine3k.pdf.

Khare, K., & Rajaratnam, B. (2011). Wishart distributions for decomposable covariance graph models. *The Annals of Statistics, 39(1)*, 514-555.

Khare, K., & Rajaratnam, B. (2012). Sparse matrix decompositions and graph characterizations. *Linear Algebra and its Applications, 437*, 932-947.

Khare, K., Oh, S., & Rajaratnam, B. (2015). A convex pseudo-likelihood framework for high dimensional partial correlation estimation with convergence guarantees. *Journal of the Royal Statistical Society Series B, 77(4)*, 803-825.

Legarra A., Robert-Granié, C., Manfredi, E., & Elsen, J.M. (2008). Performance of genomic selection in mice. *Genetics, 180*, 611-618.

Lehmann, E.L. & Casella, G. (1998). *Theory of point estimation* (2nd ed.). New York, NY: Springer.

Lenkoski, A. (2013). A Direct Sampler for G-Wishart Variates. *STAT, 2(1)*, 119-128.

Letac, G., & Massan, H. (2007). Wishart distributions for decomposable graphs. *The Annals of Statistics, 35(3),* 1278-1323.

Malovini, A., Nuzzo, A., Ferrazzi, F., Puca, A.A., & Bellazzi, R. (2009). Phenotype forecasting with SNPs data through gene-based Bayesian networks. *BMC Bioinformatics 10(2)*, S7.

Meuwissen, T.H.E., Hayes B.J., & Goddard, M.E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics, 157*,1819-1829.

Mitsakakis, N., Massam, H., & Escobar, M.D. (2011). A Metropolis-Hastings based method for sampling from the G-Wishart distribution in Gaussian graphical models. *Electronic Journal of Statistics, 5*, 18-30.

Oh, S.Y., Rajaratnam, B., & Won, J.H. (2016). Towards a sparse, scalable, and stably positive definite (inverse) covariance estimator. *arXiv preprint,* arXiv: 1502.00471 v3.

Paulsen, V.I., Power, S.C., & Smith, R.R. (1989). Schur products and matrix completions. *Journal of Functional Analysis, 85*, 151-178.

Peñagaricano, F., Weigel, K.A., Rosa, G.J.M., & Kathib, H. (2013). Inferring quantitative trait pathways associated with bull fertility from a genome-wide association study. *Frontiers in Genetics, 3*, 307.

Piccioni, M. (2000). Independence Structure of Natural Conjugate Densities to Exponential Families and the Gibbs' Sampler. *Scandinavian Journal of Statistics, 27,* 111-127.

R Core Team (2015). R: A language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria. *URL https://www.R-project.org/*.

Rajaratnam, B., Massam, H., & Carvalho, C. (2008). Flexible covariance estimation in graphical Gaussian models. *The Annals of Statistics, 36(6)*, 2818-2849.

Roverato, A. (2000). Cholesky decomposition of a hyper inverse Wishart matrix. *Biometrika*, *87(1)*, 99.112.

Roverato, A. (2002). Hyper inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian graphical models *Scandinavian Journal of Statistics,* 29, 391–411.

Sargolzaei, M., & Schenkel, F.S. (2013). *QMSim User's Guide Version 1.10*. Centre for Genetic Improvement of Livestock, Department of Animal and Poultry Science, University of Guelph, Guelph, Canada.

Scutari, M., Mackay, I., & Balding, D. (2013). Improving the efficiency of genomic selection. *Statistical Applications in Genetics and Molecular Biology, 12(4)*, 517-527.

Scutari, M., Howell, P., Balding, D.J., & Mackay, I. (2014). Multiple quantitative trait analysis using Bayesian networks. *Genetics, 198*, 129-137.

Searle, S.R., Casella, G., & McCulloch, C.E. (1992). Variance components. New York, NY: John Wiley & Sons.

Silva, R., & Ghahramani, Z. (2006). The Hidden Life of Latent Variables: Bayesian Learning with Mixed Graph Models. *Journal of Machine Learning Research, 10*, 1187-1238.

Speed, T.P., & Kiiveri, H.T. (1986). Gaussian Markov distributions over finite graphs. *The Annals of Statistics, 14(1),* 138-150.

VanRaden, P.M. (2008). Efficient methods to compute genomic predictions. *Journal of Dairy Science, 91,* 4414-4423.

Yang, W., & Tempelman, R.J. (2012). A Bayesian Antedependence Model for Whole Genome Prediction. *Genetics*, *190*, 1491-1501.

Zhang, L., Sarkar, A., & Mallick, B.K. (2013). Bayesian low rank and sparse covariance matrix decomposition. *arXiv preprint,* arXiv:1310.4195.

Zhong, S., Dekkers, J.C.M., Fernando, R.L., Jannink, J.K. (2009). Factors affecting accuracy from genomic selection in populations derived from multiple inbred lines: A barley case study. *Genetics*, 182, 355-364.

Appendix A: Basic Concepts in Graph Theory

Undirected graph. An undirected graph G is defined as a collection of two objects $G = (V, E)$ where V is the set of vertices (finite) and $E \subseteq V \times V$ is the set of edges satisfying:
$$(u, v) \in E \iff (v, u) \in E.$$
Neighbor vertices. Let $G = (V, E)$ be an undirected graph. The vertices $u, v \in V$ are said to be neighbors if $(u, v) \in E$.

P-path. A p-path is a collection of p distinct vertices $u_1, u_2, \dots, u_p$ such that $(u_i, u_{i+1}) \in E, i = 1,2, \dots, p - 1$, that is, (u_i, u_{i+1}) are neighbors for $i = 1,2, \dots, p - 1$.

P-cycle. A p-cycle is a collection of p distinct vertices $u_1, u_2, \dots, u_p$ such that $(u_i, u_{i+1}) \in E, i = 1,2, \dots, p - 1$ and $(u_p, u_1) \in E$

Clique. A subset $V_0 \subset V$ is a clique if $(u, v) \in E \; \forall \; u, v \in V_0$.

Maximal clique. A subset $V_0 \subset V$ is defined to be a maximal clique if V_0 is a clique and there does not exist a clique $\bar{V}$ such that $V_0 \subset \bar{V} \subseteq V$.

Ordered graphs. Let $G = (V, E)$ and let σ be an ordering of V, that is, a bijection from V to $\{1,2, \dots, |V|\}$. Then, the ordered graph $G_\sigma = (\{1,2, \dots, |V|\}, E_\sigma)$ is defined as follows: $(i, j) \in E_\sigma$ iff $(\sigma^{-1}(i), \sigma^{-1}, (j)) \in E$.

Perfect elimination ordering. An ordering σ of a graph $G = (V, E)$ is defined to be a perfect elimination ordering if a triplet $\{i, j, k\}$ with $i > j > k$ such that $(i, j) \notin E_\sigma$ and $(i, k), (j, k) \in E_\sigma$ does not exist.

Subgraph. The graph $G' = (V', E')$ is said to be a subgraph of graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.

Induced subgraph. Consider the graph $G = (V, E)$ and a subset $A \subseteq V$. Define $E_A = (A \times A) \cap E$. The subgraph $G_A = (A, E_A)$ is defined to be a subgraph of G induced by A. **Decomposable graph.** This is an important concept in graphical models. A graph $G = (V, E)$ is a decomposable graph if it does not contain a cycle of length greater than or equal to four as an induced subgraph. It turns out that decomposable graphs are characterized by the existence of a perfect elimination ordering of their vertices; therefore, a graph $G = (V, E)$ is decomposable iff its vertices admit a perfect elimination ordering.

Appendix B: Proof of proposition 1

Approach 1 induces either a banded or a differentially banded matrix and these structures are known to correspond to decomposable graphs. Similarly, 3.a induces block diagonal matrices which also correspond to decomposable graphs. The only non-trivial part is to prove that statements made about conditions for 3.b induce a decomposable graph. To this end, we show that it is possible to find a perfect elimination ordering, and we use the fact that the existence of a perfect elimination ordering characterizes decomposable graphs.

Proof of sufficiency

Hereinafter, consider arbitrary indices $\{i, j, k\}$ such that according to σ' $i > j > k$.

Case 1) The triplet $\{i, j, k\}$ pertains to the same block. Because each block corresponds to a complete graph, it follows that in this case it cannot be that

$$i > j > k, (i,j) \notin E_\sigma, (i,k), (j,k) \in E_\sigma \tag{A.1}$$

Case 2) $i \in l, \{j,k\} \in l', (l,l') \in \mathcal{L}$. If $(i,j) \notin E_\sigma$, by condition 1.4 it means that one of the following mutually exclusive events must occur:

- $i \notin C_{l_{l'}}$ and $j \in C_{l'_l}$
- $i \in C_{l_{l'}}$ and $j \notin C_{l'_l}$
- $i \notin C_{l_{l'}}$ and $j \notin C_{l'_l}$.

It follows that:

$$i \notin C_{l_{l'}} \Rightarrow (i,k) \notin E_\sigma$$

this suffices to show that condition $A.1$ cannot be attained under the first and third scenarios. On the other hand:

$$i \in C_{l_{l'}} \text{ and } j \notin C_{l'_l} \Rightarrow k \notin C_{l'_l} \ (\because \text{condition } 1.3)$$
$$\Rightarrow (i,k) \notin E_\sigma,$$

which is enough to show that condition $A.1$ does not hold under the second scenario.

Case 3) $(i,j) \in l, k \in l', (l,l') \in \mathcal{L}$. In this case $(i,j) \in E_\sigma$ because all blocks are complete; therefore, condition $A.1$ cannot be met.

Case 4) $i \in l, j \in l', k \in l''$. If $k \notin C_{l''_l} \cup C_{l''_{l'}}$ it is clear that $(j,k),(i,k) \notin E_\sigma \ \forall \ i \in B_l, \forall \ j \in B_{l'}$. If each one of the three blocks is linked with the other two, that is, $(l,l'),(l',l''),(l,l'') \in \mathcal{L}$, condition 1.1 implies that if all three intersection sets are non-empty and $(i,j) \notin E_\sigma$ then either $(i,k) \notin E_\sigma$ or $(j,k) \notin E_\sigma$ which prevents the occurrence of $A.1$. If one or two of the three intersection subsets is empty, by condition 1.1 it follows that $I_{l''} = \emptyset$ then:

$$k \in C_{l''_l} \Rightarrow (j,k) \notin E_\sigma$$
$$k \in C_{l''_{l'}} \Rightarrow (i,k) \notin E_\sigma,$$

Notice that this holds disregarding the location of indices j and i within their corresponding blocks; therefore, condition $A.1$ cannot be attained.

On the other hand, if one block is linked to the other two, and these are the only existent links, condition 1.2 guarantees that $A.1$ does not hold.

Until here, only linked blocks were considered. Notice that if at least one index of the triplet $\{i,j,k\}$ corresponds to a marker in an isolated (i.e., not linked) block, then $A.1$ does not happen.

In conclusion, under the four conditions of result 1 it is possible to find a perfect elimination ordering and therefore the graph induced under approach 3.b is decomposable.

Now we proceed to prove necessity of conditions 1,2 and 3. To this end, it is proven that under 3.b, if these conditions do not hold, then the induced graph is not decomposable.

Proof of necessity of conditions 1.1, 1.2 and 1.3

Hereinafter when the words "there is always" precede a condition, it is meant that under any ordering of markers, the graph always satisfies the condition.

Suppose that 1.1 does not hold. It means that:

- There is always at least one triplet of linked blocks $\{l,l',l''\}$, $1 \leq l,l',l'' \leq |\Psi|$, linked in the following way: $C_{l_{l'}} \neq C_{l_{l''}}, C_{l'_l} \neq C_{l'_{l''}}, C_{l''_l} \neq C_{l''_{l'}}$, the three pairwise intersections $I_l, I_{l'}$ and

$I_{l''}$ are non-empty, and there exist at least one triplet $\{i, j, k\}$ such that $i > j > k$ and $i \in C_{l_{l''}} \cap I_l^c, j \in C_{l'}{}_l$ or $i \in C_{l_{l'''}}, j \in C_{l'}{}_l \cap I_{l'}^c$, and $k \in I_{l''}, i > j > k$,

or

- There is always at least one triplet of linked blocks $\{l, l', l''\}$ and a triplet of indices $\{i, j, k\}$ such that $i > j > k$, exactly one or two of the three sets $\{I_l, I_{l'}, I_{l''}\}$ are empty and k pertains to a non-empty intersection set.

It is easy to notice that in any of the two events described in the first case, condition A.1 is satisfied. On the other hand, in the second case, if k pertains to the (non-empty) intersection set of a block, say I_l, then we can always pick $i \in C_{l''}{}_l$ and $j \in C_{l'}{}_l$ such that $(i, j) \notin E_\sigma$, then, because $k \in I_l$ it follows that $(i, k), (j, k) \in E_\sigma$, that is, $A.1$ holds.

Therefore, if 1.1 does not hold, a perfect elimination ordering does not exist and the graph is not decomposable. Similarly, if there is always at least one triplet of blocks $\{l, l', l''\}$ satisfying the converse of condition 1.2 then $A.1$ always holds, that is, a perfect elimination ordering does not exist and consequently the graph is not decomposable. Finally, if 1.3 does not hold then always exists at least a linked pair $\{l, l'\}$ such that $\exists\, i \in C_{l_{l'}}, j \in C_{l'}^c{}_l \cap l, k \in C_{l'}{}_l, i > j > k$ which immediately implies $A.1$.

Proof of non-necessity of condition 1.4

We refute the statement that condition 1.4 is necessary by constructing a counter-example. Suppose that a graph is comprised of two blocks. Let L_σ be the subset of E_σ containing edges comprised of one node pertaining to block 1 and one node from block 2. Also assume that block 1 contains nodes $\{a, b, c, d\}$, $C_{1_2} := \{b, c, d\}, C_{2_1} := \{w, z\}$ and $\mathcal{L} := \{(d, w), (c, w), (b, z), (c, z), (d, z)\}$. Consider any ordering satisfying the following: $\sigma(a) = 1, \sigma(b) = 2, \sigma(c) = 3, \sigma(d) = 4$. Consider $i > j > k, i \in C_{2_1}$, if $i = \sigma(w)$ then $(i, j) \notin E_\sigma \Rightarrow (i, k) \notin E_\sigma$, if $i = \sigma(z)$ then $(i, j) \notin E_\sigma \Rightarrow j = 1 = \sigma(a) \therefore \nexists\, k < j$ and consequently $A.1$ does not hold. We have found a family of perfect elimination orderings and its existence implies that the graph is decomposable even though condition 4 does not hold. It proves that condition 3 is not a necessary condition.

$\blacksquare$

Appendix C: Proof of proposition 2

Proof of sufficiency The general approach is very similar to that used to prove sufficiency of conditions stated in result 1. Let Λ be the set of blocks linked to at least one block. Assume that markers have been ordered as stated in result 2, this ordering is denoted by σ. Let B_l^σ be the set of indices corresponding to markers in block l under σ. Let $\{i, j, k\}$ be three arbitrary indices such that according to σ, $i > j > k$, and $C_l \in B_l^\sigma$, $l \in \Lambda$ be the subset of indices corresponding to markers in block l whose effects are partially correlated with the effects of a subset of markers in another block $1 \leq l \leq |\Lambda|$. In the following, it is proven that under conditions 1.4, 2.1 and 2.2 this within-block ordering strategy yields a perfect elimination ordering. Four cases depending on the position of indices i, j, k are considered.

Case 1) The triplet $\{i, j, k\}$ pertains to the same block. Because each block corresponds to a complete graph, it follows that in this case condition $A.1$ is not satisfied.

Case 2) $i \in l, \{j, k\} \in l', \{l, l'\} \in \Lambda, \rho(l) > \rho(l')$. If $(i,j) \notin E_\sigma$, by condition 1.4, it means that one of the following mutually exclusive events must occur:

- $i \notin C_l$ and $j \in C_{l'}$
- $i \notin C_l$ and $j \notin C_{l'}$
- $i \in C_l$ and $j \notin C_{l'}$

It follows that:
$$i \notin C_l \Rightarrow (i,k) \notin E_\sigma,$$
this suffices to show that condition $A.1$ cannot be attained under the first and second events. On the other hand:
$$i \in C_l \text{ and } j \notin C_{l'} \Rightarrow k \notin C_{l'} \; (\because k < j \text{ under } \sigma)$$
$$\Rightarrow (i,k) \notin E_\sigma,$$
which is enough to show that condition $A.1$ does not hold under the second scenario.

Case 3) $\{i,j\} \in l, k \in l', l > l'$. In this case $(i,j) \in E_\sigma$ because all blocks are complete; therefore, condition $A.1$ cannot be met.

Case 4) $i \in l, j \in l', k \in l'', \rho(l) > \rho(l') > \rho(l'')$. If $k \notin C_{l''}$ it is clear that $(j,k), (i,k) \notin E_\sigma \; \forall i \in B_l \; \forall j \in B_{l'}$ disregarding the position of indices i and j within their corresponding blocks and the linkage relationship among the three blocks, thus $A.1$ does not hold and, therefore; in the following, only the case $k \in C_{l''}$ is considered. If each one of the three blocks is linked with the other two, then:
$$(i,j) \notin E_\sigma \Rightarrow i \notin C_l \text{ or } j \notin C_{l'}$$
$$\Rightarrow (i,k) \notin E_\sigma \text{ or } (j,k) \notin E_\sigma \; (\because \text{ condition 2.1}).$$

Similarly, if one block is linked to the other two, and these are the only existent links, condition 2.2 guarantees that $A.1$ does not hold.

Consequently, under the two conditions of result 2 and condition 1.4, ordering σ is a perfect elimination ordering. Thus, the graph induced under approach 3.b with and edges set as defined in result 2 is decomposable.

Proof of necessity of condition 2.2

To prove necessity of condition 2.2, we show that if it is not satisfied, then σ is not a perfect elimination ordering. If there is at least one triplet of blocks $\{l, l', l''\}$ such that $(l, l'), (l, l'') \in \mathcal{L}, (l', l'') \notin \mathcal{L}, \rho(l) < \rho(l') < \rho(l'')$ then by picking $i \in C_{l''}, j \in C_{l'}$ and $k \in C_{l''}$ it follows that $i > j > k$, and $(i,j) \notin E_\sigma, (i,k), (j,k) \in E_\sigma$, that is, condition A.1 holds and as a consequence, σ is not a perfect elimination ordering, which implies necessity of condition 2.2.

Proof of non-necessity of condition 2.1

By constructing a counter example, we refute the necessity of condition 2.1. Suppose that G contains three blocks such that $\rho(B_1) = 1, \rho(B_2) = 2, \rho(B_3) = 3$, B_1 is only linked to B_2 and B_2 is only linked to B_3, $C_{2_1} \neq C_{2_3}$ and markers have been ordered according to σ. It is easy to see that in this graph, for any triplet $\{i, j, k\}$ such that according to $\sigma, i > j > k$ if $(i,j) \notin E_\sigma$ then $(i,k) \notin E_\sigma$ which implies that condition A.1 is never reached and consequently, σ is a perfect elimination ordering despite of the fact that condition 2.1 does not hold. Consequently, 2.1 is not a necessary condition.

$\blacksquare$

Appendix D: Proof of corollary to proposition 2

Let G^s be the "super graph" defined in the corollary. It is assumed that conditions 2.1 and 1.4 hold. Consequently, we only need to check that condition 2.2 is satisfied. It follows immediately by noticing that condition 2.2 is nothing but the definition of a perfect elimination ordering of the blocks, i.e., a perfect elimination ordering of the "super vertices" of G^s.

■

Appendix E: R Code

```
####################################################################
####-----Function to define connected nodes, banded structure----#####
####################################################################

Def.Graph.Band=function(band.size,nloci){
      win.size=band.size-1
       size=band.size*(nloci-band.size+1)+sum(seq(2,max(band.size-1,2),1))-nloci+1
       Connect.nodes=matrix(nrow=size,ncol=2)
      template=sort(rep(seq(1,nloci-win.size,1),win.size))
      for(i in 0:(nloci-win.size-1)){
       Connect.nodes[which(template==i+1),2]=seq(2+i,2+i+win.size-1,1)
       seq(2+i,2+i+win.size-1,1)
                                        }
      Connect.nodes[1:length(template),1]=template
      counter=seq(2,max(2,win.size-1),1)
      links=list(rep(NA,win.size-1))
      for(i in 1:(win.size-1)){
             links[[i]]=rep(nloci-i,i)
                            }
       Connect.nodes[(length(template)+1):size,1]=sort(do.call(c,links),decreasing=FAL
SE)
       links2=list(rep(NA,win.size-1))
      for(i in 1:(win.size-1)){
             links2[[i]]=seq(nloci-win.size+i+1,nloci,1)
                            }
       Connect.nodes[(length(template)+1):size,2]=do.call(c,links2)
      return(list(Connect.nodes,size))
                                            }

Nodes=Def.Graph.Band(band.size=7,nloci=nloci)[[1]]

#########################################################################
###---Function to sample from full cond. in Concentration graph models---###
#############--------------Decomposable graphs-------------------##########
#########################################################################

SampLDL=function(Delta,U,nloci,Nodes,g){
  S=tcrossprod(g)
  D=matrix(0,nrow=nloci)
  X=list(c(rep(NA,nloci-1)))
  L=diag(nloci)
  postm=S+U
  library(pscl)
  library(mvtnorm)
  library(Matrix)
   for(j in 1:(nloci-1)){
     pointer=Nodes[which(Nodes[ ,1]==j),2]
     Post1=postm[pointer,pointer]
     cov=chol2inv(chol(Post1))
```

```
      D[j]=rgamma(1,shape=(Delta+length(pointer)+2)/2,rate=(postm[j,j]-
postm[j,pointer]%*%(cov%*%postm[j,pointer]))/2)
      X[[j]]=rmvnorm(1,mean=-cov%*%postm[j,pointer],sigma=cov/D[j],method="chol")
      L[pointer,j]=X[[j]]
                              }
      D[nloci]=rgamma(1,shape=(Delta+2)/2,rate=U[nloci,nloci]/2)
      G=tcrossprod(L%*%Diagonal(length(D),sqrt(D)))
      return(list(G,matrix(do.call(c,X)),D))
                                          }

###############################################################################
###############################################################################
###########------Gibbs sampler for decomposable graphs---------################
###############################################################################
###############################################################################

BibiGraph.W=function(y,nloci,Nsim,a,U,Tau,V,W,Nodes,Burnin){

  n=nrow(W)
  X.size=nrow(Nodes)

  #####---Create arrays to save samples

  g=matrix(0,nrow=Nsim,ncol=nloci)
  norms=matrix(0,nrow=Nsim)
  gnorms=matrix(0,nrow=Nsim)
  res.var=matrix(0,nrow=Nsim)

     #####---Initial values of parameters

  library(mvtnorm)
  g[1, ]=rmvnorm(1,mean= matrix(0,nrow=nloci),sigma=diag(nloci),method="chol")

  library(pscl)
  res.var[1]=rigamma(1,alpha=V/2,beta=Tau/2)


  ############################################
  #####-----Start algorithm-----#############
  ############################################

  WTW=crossprod(W)
  WTy=crossprod(W,y)
  Ginv.hat=matrix(0,nrow=nloci,ncol=nloci)
  Ginv.sum=matrix(0,nrow=nloci,ncol=nloci)
  for(i in 2:Nsim){
    Ginv=SampLDL(Delta=a,U=U,nloci=nloci,Nodes=Nodes,g=g[i-1, ])
    norms[i]=norm(Ginv[[1]],type="F")
    cov=chol2inv(chol(Ginv[[1]] + WTW/res.var[i-1]))
    g[i, ]=rmvnorm(1, mean=crossprod(cov,WTy)/res.var[i-1],sigma=as.matrix(cov),method="chol")
    gnorms[i]=norm(matrix(g[i, ]))
    res.var[i]=rigamma(1,alpha=(V+n)/2,beta=(Tau+crossprod(y-W%*%g[i, ]))/2)
    if(i > Burnin){
              Ginv.hat=(Ginv.hat*(i-Burnin-1)+Ginv[[1]])/(i-Burnin)
                                 }
                    }

  PostmeanMar=colMeans(g[(Burnin+1):Nsim, ])

  write.table(matrix(res.var[(Burnin+1):Nsim]),col.names=FALSE,row.names=FALSE,
  file="Phen1ResVar1.csv", sep=",")

  write.table(norms[(Burnin+1):Nsim],col.names=FALSE,row.names=FALSE,
```

```
    file="Phen1norms1.csv", sep=",")

    write.table(gnorms[(Burnin+1):Nsim],col.names=FALSE,row.names=FALSE,
    file="Phen1gnorms1.csv", sep=",")

    write.table(as.matrix(Ginv.hat),col.names=FALSE,row.names=FALSE,
    file="Phen1Ginvhat1.csv", sep=",")

    write.table(PostmeanMar,col.names=FALSE,row.names=FALSE,
    file="Phen1PostmeanMark1.csv", sep=",")

    return(list(PostmeanMar,g[(Burnin+1):Nsim,
],res.var[Burnin:Nsim],norms[(Burnin+1):Nsim]))


                                                            }
U=round(as.matrix(SampLDL(Delta=7,U=diag(nloci),nloci=nloci,Nodes=Nodes,g=rnorm(nloci,
0,1))[[1]]),2)+diag(nloci)

Test=BibiGraph.W(y=data1.Training$Phen,nloci=nloci,Nsim=25000,a=7,U=U,Tau=230,V=4.1,W=
W.Train,Nodes=Nodes,Burnin=10000)

#--Predicted Breeding values
W.Test=W[which(data1$G==3), ]
PredBVTrainGEMB=W.Train%*%as.matrix(Test[[1]])
PredBVTestGEMB=W.Test%*%as.matrix(Test[[1]])


#################################################
###--------Predictive abilities---------------####
#################################################

Predabil.GB=cor(data1.Test$Phen,PredBVTestGEMB,method="pearson")

#########################################
###--correlation of breeding values---###
#########################################

corrBVGB=cor(data1.Test$QTL,PredBVTestGEMB,method="pearson")

###########################################################
###---Correlations of breeding values in training----####
###########################################################

corrBVTrainGB=cor(data1.Training$QTL,PredBVTrainGEMB,method="pearson")

Performance=matrix(c(Predabil.GB,corrBVGB,corrBVTrainGB))
write.table(Performance,col.names=FALSE,row.names=FALSE,file="Bayesperform1.csv",
sep=",")




##############################################################################
######-----Function to obtain MLE of Omega for decomposable graphs------#########
##############################################################################

clsdfrmlefordecompG = function(S,adjmat){
(L = S)
(sqD = S)
(D = S)
L[] = 0
sqD[] = 0
D[] = 0
```

```
(p = dim(L)[1])

for (i in 1:p){(Snew = S[i:p,i:p])
                if(i<p){(newp = dim(Snew)[1])
                        (Sii = Snew[1,1])
                        (temp = which(adjmat[(i+1):p,i]>0))
                        (dimtemp = length(temp))
                        if(dimtemp>0){
                        (Stempi = S[(i+1):p,i])
                          (Sdoti = Stempi[temp])
                          (Stempgp = S[(i+1):p,(i+1):p])
                          if(length(Stempgp)>1){
                                (Sgp = Stempgp[temp,temp])}
                          else {Sgp = Stempgp}
                        (xi = -solve(Sgp)%*%Sdoti)

                         (dii = sqrt(1/(Sii + t(Sdoti)%*%xi)))

                        (L[i,i] = 1)
                        (L[(temp+i),i] = xi)
                        (sqD[i,i] = dii)
                        (D[i,i] = dii^2)}
                        else{
                        (dii = sqrt(1/(Sii)))
                        (L[i,i] = 1)
                        (L[(temp+i),i] = 0)
                        (sqD[i,i] = dii)
                        (D[i,i] = dii^2)}
                  }
                else {dii = sqrt(1/Snew)
                      L[i,i] = 1
                      sqD[i,i] = dii
                      D[i,i] = dii^2}}
if(missing(adjmat)){
    cat('Error!Missing Adjacency Matrix!')
    return(0)}
return((L%*%D)%*%t(L))}

####--Function to create the adjacency matrix---###
##--This version builds a symmetric matrix instead of an upper triangular one--##

Adj2=function(Nodes,nloci){
      library(Matrix)
      Adj=as(diag(nloci),"dgTMatrix")
     for(i in 1:nrow(Nodes)){
            Adj[Nodes[i,1],Nodes[i,2]]=1
            Adj[Nodes[i,2],Nodes[i,1]]=1
                            }
     return(Adj)
                    }

Adjo=Adj2(Nodes=Nodes,nloci=nloci)

#############################################################################
#############################################################################
############----------EM algorithm for decomposable graphs-------------#########
#############################################################################
#############################################################################

Bibi.GEM=function(y,W,Adjo,Omega0,res.var0,epsilon=1e-04,Maxiter=100,template,sizes){

      N=length(y)
      f=max(template[ ,3])
```

```
    nloci=ncol(W)
     iter=0
    SIGMA0=chol2inv(chol(as.matrix(Omega0)))
    temp0=(W%*%SIGMA0)%*%t(W)+(res.var0*diag(N))
     Omega=as.matrix(Omega0)
     res.var=res.var0
     delta=epsilon*2

   while(epsilon<delta && iter< Maxiter){
    Oldres.var = res.var
    OldOmega=Omega
     Sg.sum=matrix(0,nrow=nloci,ncol=nloci)
     trace=0
    inner=0
     SIGMA=chol2inv(chol(Omega))
     Block=list(rep(NA,f))

###---Expectation step

    for(i in 1:f){
         indices = template[which(template[,3]==i),1]
         WindSig = W[indices, ]%*%SIGMA
          Vinv=chol2inv(chol((WindSig)%*%t(W[indices, ])+(diag(sizes[i])*res.var)))
          Block[[i]]=Vinv
          summand=SIGMA%*%(diag(nloci)- (t(W[indices, ])%*%(Vinv-
crossprod(t(y[indices])%*%Vinv)))%*%WindSig)
          Sg.sum=Sg.sum+summand
                 }

    Sg.expect=Sg.sum/f
    Vin=as.matrix(bdiag(Block))
    inner.res.expect=as.numeric(res.var*(N-res.var*(sum(diag(Vin))-
crossprod(Vin%*%y))))

##---Maximization step

    res.var=inner.res.expect/N
    Omega=clsdfrmlefordecompG(S=Sg.expect,adjmat=Adjo)

##---Evaluate convergence

    delta=mean(c(mean(abs(Omega-OldOmega)),abs(Oldres.var-res.var)))

    iter=iter+1


                                   } #end of the while loop

    if(delta>epsilon){print("Convergence criteria was not met")}


    write.table(Omega,col.names=FALSE,row.names=FALSE,file="EM_OmegaHat1.csv",
sep=",")
    write.table(res.var,col.names=FALSE,row.names=FALSE,file="EM_res.varHat1.csv",
sep=",")

    return(list(Omega,res.var,iter,delta))

                                        }

###--Sizes is a file containing the number of individuals per family, in this ###--
case, number of progeny per sire.
###--template is a matrix containing three columns and as many rows as
```

```
###---observations. The first column corresponds to the positions (row numbers)
###---of data corresponding to each group, the second one is the original group
###---number and the third one a consecutive group number going from 1 to f,
###---where f is the number of groups. In this case f=number of sires.

Run=Bibi.GEM(y=data1.Training$Phen,W=W.Train,Adjo=Adjo,Omega0=Omega0,res.var0=44,epsil
on=0.0001,Maxiter=1000,template=template,sizes=sizes)
OMEGAB=Run[[1]]

################################################################################
################################################################################
####--------------------SOLVING MME TO GET g_hat--------------------#########
################################################################################
################################################################################

Solve.MME=function(y,W,OMEGA,Resvar){
      RHS=crossprod(W,y)
      LHS=crossprod(W)+Resvar*OMEGA
      ghat=chol2inv(chol(LHS))%*%RHS
      return(ghat)
                                   }

ghat=Solve.MME(y=data1.Training$Phen,W=W.Train,OMEGA=OMEGAB,Resvar=Run[[2]])

#--Predicted Breeding values
W.Test=W[which(data1$G==3), ]
PredBVTrainGEMB=W.Train%*%ghat
PredBVTestGEMB=W.Test%*%ghat

##################################################
###--------Predictive abilities---------------####
##################################################

Predabil.GB=cor(data1.Test$Phen,PredBVTestGEMB,method="pearson")

#########################################
###--correlation of breeding values---###
#########################################

corrBVGB=cor(data1.Test$QTL,PredBVTestGEMB,method="pearson")

########################################################
###---Correlations of breeding values in training----####
########################################################

corrBVTrainGB=cor(data1.Training$QTL,PredBVTrainGEMB,method="pearson")

Performance=matrix(c(Predabil.GB,corrBVGB,corrBVTrainGB))
write.table(Performance,col.names=FALSE,row.names=FALSE,file="GEMperform1.csv",
sep=",")
```