Running head: Single-step genomic BLUP

Single-step Genomic Evaluation

Daniela Lourenco[1], Andres Legarra[2], and Ignacy Misztal[1]

[1]Department of Animal and Dairy Science, University of Georgia, Athens, GA, USA

‡ Institut National de la Recherche Agronomique, UMR1388 GenPhySE, Castanet Tolosan, France 31326

Email addresses:

DL: danilino@uga.edu

AL: andres.legarra@inra.fr

IM: ignacy@uga.edu

2019

Abstract

Single-step genomic evaluation became a standard procedure in livestock breeding, and the main reason is the ability to combine all pedigree, phenotypes, and genotypes available into one single evaluation, without the need of post-analysis processing. Therefore, the incorporation of data on genotyped and non-genotyped animals in this method is straightforward. Since 2009, two main implementations of single-step were proposed. One is called single-step genomic BLUP (ssGBLUP) and uses single nucleotide polymorphisms (SNP) to construct the genomic relationship matrix; the other is the single-step Bayesian regression (ssBR) which is a marker effect model. Under the same assumptions, both models are equivalent. The implementation of ssGBLUP into the BLUPF90 software suite was done in 2009, and since then, several changes were made to make ssGBLUP flexible to any model, number of traits, amount of phenotypes, and number of genotyped animals. Single-step GBLUP from BLUPF90 has been used for genomic evaluation worldwide. In this chapter we will show theoretical developments and numerical examples of ssGBLUP from the BLUPF90 software suite.

Key words: Genomic selection, single-step genomic BLUP, BLUPF90, genomic relationship matrix

1 Introduction

In the early 1980's, Soller, Beckmann [1] hypothesized that DNA markers like RFLP (i.e., restriction fragment length polymorphisms) would be beneficial in constructing more precise genetic relationships, followed by parentage determination, and the identification of quantitative trait loci (QTL). The high cost of genotyping animals for such markers probably

prevented the early widespread use of this technology. When the first draft of the Human Genome Project became available in 2001 [2], one of the most exciting news that came along was that the majority of the genome sequence variation can be attributed to single nucleotide polymorphisms (SNP). The reality is that SNP markers have become the bread-and-butter of DNA sequence variation [3] and they are now an important tool to determine the genetic potential of livestock. This is because SNPs are abundant, as they are found throughout the entire genome [4], as in introns, exons, promoters, enhancers, or intergenic regions. In fact, there are about 3 billion nucleotides in the bovine genome, and there are over 30 million SNPs or 1 every 100 nucleotides is a SNP. Another reason is that SNP genotyping became relatively cheap.

In 2001, Meuwissen et al. [5] envisioned that genomic information could help animal breeders to generate more accurate breeding values, if a dense SNP assay that covers the entire genome become available. Extending the idea of incorporating marker information into BLUP, introduced by Fernando, Grossman [6], Meuwissen et al. [5] proposed what is now termed genome-wide selection or genomic selection (GS). The Bayesian models described in Meuwissen et al. [5] provide SNP effects and direct genomic values (DGV) based on genotypes and phenotypes or pseudo-phenotypes (i.e., EBV adjusted for parent average and accuracy) only for genotyped animals. Following the same line, VanRaden [7] proposed a method called GBLUP, where predictions for genotyped animals are obtained based on genomic relationships (i.e., proportion of alleles shared between animals) instead of pedigree relationships. This genomic relationship matrix is represented by **G**. After using GBLUP or Bayesian methods, a post-processing step is needed to account for pedigree information; therefore, the traditional BLUP evaluation is still needed. Because several steps are needed to retrieve genomic EBV (GEBV), this class of methods is called multistep. The main advantage of this approach is that

the traditional BLUP evaluation is kept unchanged and genomic selection can be carried out by using additional analyses. However, this method has some disadvantages: a) DGV are only generated for simple models (i.e., single trait, non-maternal models), which is not the reality of genetic evaluations; b) only genotyped animals are included in the model; c) it requires pseudo-phenotypes that rely on accuracy obtained via approximated algorithms.

Although multistep methods were largely implemented for genomic evaluations worldwide, starting from 2009, it seems this class of methods was never intended to be the enduring process to compute genomic predictions. This is because only a fraction of pedigreed animals is genotyped and the genomic information cannot be extended to non-genotyped animals; therefore, genotyped animals have GEBV and non-genotyped have EBV. As a result, several adjustments were proposed, especially in dairy cattle, to make EBV comparable to GEBV under multistep evaluations [8,9]. Intending to solve incompatibility problems and to reduce the burden in obtaining genomic predictions, Misztal et al. [10] proposed a method that combines phenotypes, pedigree, and genotypes into a single evaluation. This method is called single-step genomic BLUP (ssGBLUP) and involves replacing the pedigree relationship matrix in the traditional BLUP by a realized relationship matrix, which combines pedigree and genomic relationships. This realized relationship matrix was named **H** matrix. If the question is why **H**, the answer is quite simple: if the genomic relationship is represented by **G**, just pick the next letter in the alphabet.

Still in 2009, Legarra et al. [11] showed that the pedigree relationship can be viewed as *a priori* relationship and the genomic relationship as the observed relationship. The derivation of the joint distribution of pedigree and genomic relationships would allow the extension (or imputation) of genomic information to non-genotyped animals. This means that in ssGBLUP

pedigree relationships for non-genotyped animals are enhanced by the genomic information of their relatives. After 10 years, ssGBLUP has become the preferred tool for genomic evaluation and selection in many livestock species, namely Beef cattle[12], pigs [13,14], broilers [15,16], layers [17], dairy sheep and goat [18], meat sheep [19], and fish [20]. Although ssGBLUP adds simplicity to the genomic evaluation system, its implementation involves several details and requires knowledge about peculiarities of the method. In this chapter we will show theoretical developments and numerical examples of ssGBLUP, from the BLUPF90 software suite, that will ease the steps toward the application of the method.

2 Software, Methods, and Algorithms

2.1 BLUPF90 software suite

BLUPF90 is a collection of software for computations with focus on applications in breeding and genetics. It is based on Fortran 90/95 and started being developed in 1999 by Ignacy Misztal, with the objective to be simple and flexible for model fitting. The first idea was to have a simple BLUP program to compute solutions for the mixed model equations (MME), then `blupf90` was the first software created. This software supports general multiple-trait models, different model design per trait, multiple effects, missing data, random correlated and non-correlated effects, dominance effects, and can use several pedigree files or different covariance structures defined by the user [21].

After the first software (i.e., `blupf90`), several programs were developed to support variance components estimation for linear models (i.e., `remlf90, airemlf90, gibbsf90`) and linear-threshold models (`thrgibbsf90`), accuracy approximation (`accf90`),

and large-scale genetic evaluations using linear models (`blup90iod2`) and linear-threshold models (`cblup90iod`). For information on how to download and use the programs, check Note 1.

Additionally, a renumbering program (i.e., `renumf90`) was created that also provides data statistics, performs extensive pedigree checks, can assign unknown parent groups (UPG), supports large data sets, and creates a parameter file that can be used as input for all software in the BLUPF90 suite (see Note 2).

When genomic information became available and ssGBLUP was developed, the flexibility of the BLUPF90 family of programs allowed the efficient incorporation of genomics. The extra file with gene content for each animal is easily read, then genomic relationships are computed and can be used by any software in the family. This is because all the programs share the same genomic library, which contains all functions to deal with genomic data. Additionally, a software was developed (i.e., `pregsf90`) to perform quality control and preprocessing of genomic data, and to be the main interface to the genomic library (see Note 3). All the programs, but the ones for large-scale evaluations, are freely available for research and academic purposes. Linux, Windows, and Mac versions can be downloaded here: http://nce.ads.uga.edu/html/projects/programs. General descriptions about all the programs are available here http://nce.ads.uga.edu/wiki/doku.php?id=application_programs. The current free software can handle up to 25,000 genotypes animals; however, this limitation bar is frequently raised. Additionally, all software in the BLUPF90 family is under constant development and new updates are released several times a year.

2.2 Genomic relationship-based methods

Single-step GBLUP is considered a genomic relationship-based method. This class of methods use SNP to infer relationships among individuals, quantifying the number of alleles shared between two individuals. Genomic relationships are identical by state (IBS) because they account for the probability that two alleles randomly picked from each individual are identical, independently of origin. Pedigree relationships are identical by descent (IBD) because they consider the shared alleles come from the same ancestor.

Assuming a matrix of SNP inherited by each animal (**Z***), with dimension *n* x *m* where *n* is the number of animals and *m* the number of SNP. Several parametrizations exist, but if AA=0, AB=1, and BB=2, **Z*** has to be centered by allele frequency. Assuming a matrix **P** with elements equal to $2p_i$, with p_i being the minor frequency of allele *i*:

$$\mathbf{Z} = \mathbf{Z}^{*} - \mathbf{P} \quad (1)$$

To understand why **Z** is a centered matrix of allele content, we can use only one biallelic marker. If the effect of each copy of the A allele is *a* and the frequency of AA is p^2, individuals with AA have a breeding value u = *2a*; individuals aa have u = *0* with a frequency of q^2; individuals Aa have u = *a* with a frequency *2pq*. The variance explained by this marker is $Var(\mathrm{u}) = \mathrm{E}(\mathrm{u}^2) - E(\mathrm{u})^2$ [22]. The average of *u* is $2ap^2 + a2pq$; which becomes *2pa*. The variance explained by one marker is:

$(2a)^2p^2 + 2pq(a)^2 - (2pa)^2 = 2pqa^2$. Given the average of u is *2pa*, as shown above, we can compute the covariance between individuals *i* and *j* for this marker. If we express the breeding values of the animals *i* and *j* as *za* deviated from the population mean [22]:

$$\mathrm{u_i} = z_i a - 2pa = (z_i - 2p)a \quad (2)$$

$$\mathrm{u_j} = z_j a - 2pa = (z_j - 2p)a \quad (3)$$

According to Legarra et al. [22], if $Var(a) = \sigma_a^2$, or marker variance, and the genetic variance in Hardy-Weinberg equilibrium is $2pq\sigma_a^2$, the rules of variances and covariances can be applied:

$$Cov(u_i, u_j) = (z_i - 2p)a(z_j - 2p)a = (z_i - 2p)(z_j - 2p)\sigma_a^2 \quad (4)$$

If instead of using the allele coding 0,1,2 we use -1,0,1:

$$Cov(u_i, u_j) = z_i z_j \sigma_a^2 \quad (5)$$

Dividing the covariance by the genetic variance $2pq\sigma_a^2$, we get realized relationships.

Going from one to several markers, the breeding value of an animal can be calculated as the sum of SNP effects weighted by the genotype content ($\mathbf{u} = \mathbf{Za}$). Assuming the same variance per locus, the variance of $\mathbf{u}$ is:

$$Var(\mathbf{u}) = Var(\mathbf{Za}) \quad (6)$$

$$Var(\mathbf{u}) = \mathbf{Z}Var(\mathbf{a})\mathbf{Z}' \quad (7)$$

$$Var(\mathbf{u}) = \mathbf{ZZ}'\sigma_a^2 \quad (8)$$

If the genetic variance $\sigma_u^2 = 2\sum_{i=1}^{SNP} p_i(1 - p_i)\,\sigma_a^2$, then $\sigma_a^2 = \frac{\sigma_u^2}{2\sum_{i=1}^{SNP} p_i(1-p_i)}$. Replacing σ_a^2 in (8) we have that:

$$Var(\mathbf{u}) = \mathbf{ZZ}'\frac{\sigma_u^2}{2\sum_{i=1}^{SNP} p_i(1-p_i)} \quad (9)$$

$$Var(\mathbf{u}) = \frac{\mathbf{ZZ}'}{2\sum_{i=1}^{SNP} p_i(1-p_i)}\sigma_{\boldsymbol{u}}^2 \quad (10)$$

According to VanRaden [7], the genomic relationship (**G**) is given by:

$$\mathbf{G} = \frac{\mathbf{ZZ}'}{2\sum p_i(1-p_i)} \quad (11)$$

$$\text{then, } Var(\mathbf{u}) = \mathbf{G}\sigma_u^2 \quad (12)$$

Therefore, genomic relationships are standardized covariances. When $\mathbf{ZZ}'$ is divided by $\sum p_i(1-p_i)$, **G** becomes analogous to the numerator relationship matrix (**A**). The **G** matrix contains the number of homozygous loci for each individual in the diagonals, and the number of alleles shared among individuals in the off-diagonals. Other ways to construct the genomic relationship matrix are described in the literature. For more details, check Leutenegger et al. [23] and Amin et al. [24].

If **G** is centered using current allele frequencies, the average over all elements is zero and average diagonal is 1 when there is no inbreeding. In general, **G** traces inbreeding much further than **A** because of its IBS nature and because **A** is limited by the recent pedigree recording.

When the number of genotyped animals is bigger than the number of SNP, or if there are similar individuals (e.g., clones), **G** becomes singular, therefore, cannot be inverted. To overcome this problem, **G** is modified to have larger diagonals. Usually, **G** is blended with 1% or 5% of an identity matrix or the pedigree relationship matrix among genotyped animals ($\mathbf{A}_{22}$):

$$\mathbf{G} = \alpha\,\mathbf{G} + (1-\alpha)\,\mathbf{A}_{22} \qquad (13)$$

2.3 From GBLUP to ssGBLUP

Understanding the difference between GBLUP and ssGBLUP is a crucial step in this chapter. Because there is still a lot of confusion, an explanation about GBLUP is provided.

The GBLUP is equivalent to SNP-BLUP, but genomic breeding values ($\mathbf{u} = \mathbf{Za}$) are estimated instead of SNP effects (**a**). It also assumes that SNP explain the same proportion of variance; however, the majority of SNP have a small effect and very few have moderate to large effect. Using a simple animal model:

$$\mathrm{y} = \mathbf{X}b + \mathbf{W}u + e \qquad (14)$$

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'W} \\ \mathbf{W'X} & \mathbf{W'W}+\mathbf{G}^{-1}\frac{\sigma_e^2}{\sigma_u^2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{W'y} \end{bmatrix} \qquad (15)$$

where **W** is the incidence matrix for animal effect (**u**), **X** is the incidence matrix for fixed effects (**b**), σ_e^2 is the residual variance, and $\mathbf{u} \sim N(0, \mathbf{G}\sigma_u^2)$.

Therefore, GBLUP is a BLUP where the pedigree relationship matrix (**A**) is replaced by the genomic relationship matrix. The effectiveness of GBLUP will depend on the ability of **G** to approach the realized genetic relationships. In addition, performing a quality control of genomic data before constructing **G** avoids biases and losses of accuracy.

If we assume that not all the genetic variance is explained by markers, an extra polygenic effect can be included to explain the remaining variance. In this case, the model in (14) becomes:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Wu} + \mathbf{Wg} + \mathbf{e} \qquad (16)$$

where g is a vector of residual polygenic effect that is not captured by the SNPs. Assuming that α is the proportion of variance explained by SNPs, the total additive genetic effect (u_g) becomes

$$\mathbf{u}_\mathrm{g} = \mathbf{u} + \mathbf{g} \qquad (17)$$

$$Var(\mathbf{u}_\mathrm{g}) = \alpha\mathbf{G}\sigma_g^2 + (1-\alpha)\mathbf{A}_{22}\sigma_u^2 \qquad (18)$$

$$\text{Therefore, } \mathbf{G} = \alpha\mathbf{G} + (1-\alpha)\mathbf{A}_{22} \qquad (19)$$

In real situations, it is assumed that α varies from 0.8 to 0.95. Note that this is also going to make **G** invertible [25]. When α is used strictly to make **G** (semi-) positive definite, it is called blending parameter.

Although GBLUP has been widely used in animal and plant breeding applications, its main problem is that only genotyped animals are in the model. As only a fraction of animals is genotyped, GBLUP may have less phenotypic and pedigree information than BLUP. Because of that, some extra steps are needed to combine genomic and pedigree information. When using GBLUP, SNP-BLUP or Bayesian models, the genomic evaluation method is called multistep. The steps involved in multistep are: 1) estimation of EBV using traditional BLUP (i.e., all available information); 2) de-regression of EBV, which condenses information from phenotypes (e.g., daughter yield deviation in dairy cattle); 3) estimation of SNP effects using GBLUP or other models; 4) prediction of **Za**, which is also known as direct genomic values (DGV); 5) blending DGV with average of parent's EBV, which is known as parent average (PA). The main issue on having an evaluation with several steps is that some errors and biases can be introduced during those steps [26].

The idea for ssGBLUP came from the fact that only a small portion of the animals, in a given population, is genotyped. In this way, the best approach to avoid several steps would be to combine pedigree and genomic relationships and use this matrix as the covariance structure in the MME. Legarra et al. [11] stated that genomic evaluations would be simpler if genomic relationships were available for all animals in the model. Then, their idea was to look at **A** as *a priori* relationship and to **G** as observed relationships; however, **G** is observed only for some individuals, and those individuals have $\mathbf{A}_{22}$ as *a priori* relationship. Based on that, it was shown that the genomic information could be extended to non-genotyped animal based on the joint distribution of breeding values of non-genotyped ($\mathbf{u}_1$) and genotyped ($\mathbf{u}_2$) animals [25,11]:

$$p(\mathbf{u}_1, \mathbf{u}_2) = p(\mathbf{u}_2)p(\mathbf{u}_1|\mathbf{u}_2) \qquad (20)$$

$$p(\mathbf{u}_2) = N(0, \mathbf{G}) \qquad (21)$$

If we consider that

$$var(\mathbf{u}) = \mathbf{A}\sigma_u^2 \qquad (22)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \qquad (23)$$

where subscripts 1 and 2 represent non-genotyped and genotyped animals, respectively. The conditional distribution of breeding values for non-genotyped and genotyped animals is

$$p(\mathbf{u}_1|\mathbf{u}_2) = N(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}u_2, \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}) \qquad (24)$$

If $\mathbf{u}_2$ in $\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2$ is replaced by a vector of observed gene content, the formula can be used to estimated gene content for non-genotyped animals based on observed gene content for genotyped animals [27]. It implies that by using $\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2$, the genomic information can be implicitly imputed from genotyped animals to non-genotyped based on pedigree relationships. The variance in the distribution ($\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$) is the prediction error term. Therefore, because the animals with subscript 1 have no genotypes, the variance depends on their pedigree relationships with genotyped animals. In this way, variances and covariances are:

$$var(\mathbf{u}_1) = var(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2 + \varepsilon) \qquad (25)$$

$$= var(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2) + var(\varepsilon)$$

$$= \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} + \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$$

Rearranging:

$$= \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$$

$$= \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{I}\mathbf{A}_{21}$$

$$var(\mathbf{u}_1) = \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{22}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$$

Therefore,

$$var(\mathbf{u}_1) = \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{G} - \mathbf{A}_{22})\mathbf{A}_{22}^{-1}\mathbf{A}_{21} \qquad (26)$$

$$var(\mathbf{u}_2) = var(\mathbf{Za}) = \mathbf{G} \qquad (27)$$

$$cov(\mathbf{u}_1, \mathbf{u}_2) = cov(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2, \mathbf{u}_2) \qquad (28)$$

$$= \mathbf{A}_{12}\mathbf{A}_{22}^{-1}var(\mathbf{u}_2)$$

$$cov(\mathbf{u}_1, \mathbf{u}_2) = \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G} \qquad (29)$$

Finally, the matrix that contains the joint relationships of genotyped and non-genotyped animals is given by:

$$\mathbf{H} = \begin{pmatrix} var(\mathbf{u}_1) & cov(\mathbf{u}_1, \mathbf{u}_2) \\ cov(\mathbf{u}_2, \mathbf{u}_1) & var(\mathbf{u}_2) \end{pmatrix} \qquad (30)$$

$$= \begin{pmatrix} \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{G} - \mathbf{A}_{22})\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G} \\ \mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{G} \end{pmatrix}$$

$$\mathbf{H} = \mathbf{A} + \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{G} - \mathbf{A}_{22})\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{G} - \mathbf{A}_{22}) \\ (\mathbf{G} - \mathbf{A}_{22})\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{G} - \mathbf{A}_{22} \end{bmatrix} \qquad (31)$$

which can be simplified to:

$$\mathbf{H} = \mathbf{A} + \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} [\mathbf{G} - \mathbf{A}_{22}] [\mathbf{I} \quad \mathbf{I}] \begin{bmatrix} \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \qquad (31)$$

This **H** matrix is, therefore, a relationship matrix constructed with SNP markers and pedigree, where the SNP information is projected to the individuals that are not genotyped. Some of its properties include: being always semi-positive definite, and being positive definite and

invertible if **G** is invertible. Although **H** is very complicated, its inverse ($\mathbf{H}^{-1}$) is quite simple [28,25] :

$$\mathbf{H}^{-1}=\mathbf{A}^{-1}+\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{G}^{-1}-\mathbf{A}_{22}^{-1}\end{bmatrix} \qquad (32)$$

As both $\mathbf{A}^{-1}$ and $\mathbf{G}^{-1}$ capture relationships, $\mathbf{A}_{22}^{-1}$ should be subtracted to avoid double-counting of pedigree information for genotyped animals.

Assuming the following animal model:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Wu} + \boldsymbol{e} \qquad (33)$$

The MME for ssGBLUP becomes:

$$\begin{bmatrix}\mathbf{X'X} & \mathbf{X'W}\\ \mathbf{W'X} & \mathbf{W'W}+\mathbf{H^{-1}}\frac{\sigma_e^2}{\sigma_u^2}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{b}}\\ \hat{\mathbf{u}}\end{bmatrix}=\begin{bmatrix}\mathbf{X'}\mathrm{y}\\ \mathbf{W'}\mathrm{y}\end{bmatrix} \qquad (34)$$

The distribution of **u** becomes:

$$\mathbf{u}\sim N(0,\mathbf{H}\sigma_u^2) \qquad (35)$$

Therefore, the only difference between BLUP and ssGBLUP is that **A** is replaced by **H**. Subsequently, all tools based on BLUP MME, as the restricted maximum likelihood (REML [29]), can be converted to single-step. In a nutshell, if all animals are genotyped, ssGBLUP becomes GBLUP, but if no animals are genotyped, ssGBLUP becomes BLUP.

Advantages of ssGBLUP include simplicity of use, simultaneous fit of genomic information and estimation of fixed effects [26], relatively higher accuracy than multistep methods [30-34], potential to account for pre-selection bias as all pedigree, phenotypic, and genomic information can be included in the model [35,9], can be easily extended to any model.

2.4 Applying ssGBLUP to a simulated data using `blupf90`

A dataset that mimicked a cattle population was simulated using QMSim [36]. Pedigree information and phenotypes for 10,000 animals, and genotypes for 1020 parents from generations 1-4 and 1004 individuals in generation 5 were generated. Files with pedigree, phenotypes, and genotypes are available at https://github.com/danielall/Data_ssGBLUP. Shortly, the pedigree file is named pedigree.txt and contains 3 columns: animal, sire, dam. The phenotype file is named phenotypes.txt and contains: animal, sex, phenotype, true breeding value, generation. Phenotypes (y) were generated as y = sex_effect + true_breeding_value + residual. Genotypes were coded based on the number of copies of the alternative allele (0, 1, 2) and are in a file named genotypes.txt, with: animal, SNP_genotype. The last file (gen_map.txt) contains the map for SNP: SNP order, chromosome number, position (bp).

After running `renumf90` to renumber the data (see Note 2), the renumbered phenotype file is named renf90.dat and contains phenotype, renumbered sex code, and renumbered animal ID; the renumbered pedigree file is renadd02.ped; and the parameter file generated by `renumf90` is named renf90.par (Box 1). This parameter file was created based on the following model: y = sex + u + residual, where u is the animal effect or direct additive genetic effect. To run ssGBLUP, `blupf90` can be used with the parameter file given in Box 1 (see Note 2 for a description of keywords and values). The following command line can be used to save the screen output to a file: `echo` renf90.par | `blupf90` | `tee` blupout.log

The above command will provide the parameter file when `blupf90` asks for it and will save the screen output to a file named blupout.log

Box 1: Parameter file for running ssGBLUP in `blupf90`

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
      1
NUMBER_OF_EFFECTS
      2
OBSERVATION(S)
   1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE
NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT
NESTED]
 2       2 cross
 3    12010 cross
RANDOM_RESIDUAL VALUES
  0.60000
 RANDOM_GROUP
    2
 RANDOM_TYPE
 add_an_upginb
 FILE
renadd02.ped
(CO)VARIANCES
  0.40000
OPTION SNP_file genotypes.txt
OPTION chrinfo gen_map.txt
```

Preconditioner conjugate gradient [37] is the default method used by `blupf90` to solve the MME; however, other options exist. To check all options `blupf90` can take, check this link: http://nce.ads.uga.edu/wiki/doku.php?id=readme.blupf90

The only output file provided by `blupf90` is a solution file named “solutions”, and the first 5 lines of this file are shown in Box 2. The first line is a header indicating columns for trait, effect, level, and solution. In this example, only one trait was used, so all entries in the trait

column are 1; the effect column contains the number of the effects in the model, i.e., sex and animal effect; level refers to the levels of the effects, i.e., 2 for sex and 12,010 for animal effect (direct additive genetic); the last column contains the solutions for all levels of the effects in the model. As ssGBLUP was used by `blupf90` because the option `OPTION SNP_file` was included, solutions of the animal effect are GEBV for both genotyped and non-genotyped animals. It is important to remember the effects were renumbered using `renumf90`, so the original and renumbered levels for fixed effects and animal effect are in renf90.tables and renadd02.ped, respectively (see Note 2).

Box 2: First 5 lines of `blupf90` solutions file

trait	effect	level	solution
1	1	1	2.43346240
1	1	2	1.44508009
1	2	1	0.05317279
1	2	2	-0.16549683

To have GEBV matched back to the original ID, a simple R script, as the one in Box 3, can be used.

Box 3: Merging GEBV with original animal ID

```
rm(list=ls())
sol<-read.table("solutions", skip=1)
sol_gebv<-subset(sol,sol[,2]==2)
names(sol_gebv)<-list("trait","effect","level","solutions")
ped<-read.table("renadd02.ped")
ids<-data.frame(ped[,1],ped[,10])
names(ids)<-list("level","orig_level")
sol_orig_id<-merge(ids,sol_gebv,by="level")
write.table(sol_orig_id,file="sol_orig_id.txt",quote=F,row.names=F)
```

Although solutions is the only file generated by `blupf90`, the software outputs a large amount of information on the screen, including quality control checks, statistics for **G** and $\mathbf{A}_{22}$ and respective inverses, and statistics for $\mathbf{G}^{-1}$ - $\mathbf{A}_{22}^{-1}$. This is because when the option `OPTION SNP_file` is used in `blupf90`, it turns the genomic library on and all checks are done. To avoid doing quality control of genomic data when using `blupf90`, add the following option at the end of the parameter file: `OPTION no_quality_control`. The genomic library has an interface software called `preGSf90`, which contains a myriad of options. To check all options available in the genomic library: http://nce.ads.uga.edu/wiki/doku.php?id=readme.pregsf90. To see how to use `preGSf90` to perform quality control and preprocessing of genomic data, check Note 3.

2.5 Compatibility between pedigree and genomic relationships

Based on how **H** is constructed, the central element is $\mathbf{G} - \mathbf{A}_{22}$, which implies both matrices should be compatible [26,38]. However, genomic relationships can be biased if **G** is constructed based on allele frequencies other than the ones calculated from the base population [7]. Allele frequencies from the base population are not known because of the recent recording of pedigrees (i.e., the base population *per se* is unknown). Although those frequencies can be estimated using the method proposed by Gengler et al. [27], the computing cost is quite high because it requires to solve the BLUP MME for each SNP marker. Most commonly, allele frequencies based on the recent genotyped population are used to construct **G**. When this is the case, the expectation of breeding values for genotyped animals is 0 [7]. However, if the population is under selection, mean breeding values can deviate from 0. To account for selection

and for the fact genotyped animals are more related through $\mathbf{A}_{22}$ than **G** is able to reflect (i.e., especially when current allele frequencies are used), Vitezica et al. [38] proposed an adjustment factor (ρ) to match averages of **G** to averages of $\mathbf{A}_{22}$. This adjustment was crucial to avoid bias in ssGBLUP evaluations, especially in populations is under selection. It can be calculated as:

$$\rho = \frac{1}{n^2}\left(\sum_i \sum_j \mathbf{A}_{22\,i,j} - \sum_i \sum_j \mathbf{G}_{i,j}\right) \qquad (36)$$

where n is the number of elements in $\mathbf{A}_{22}$ and **G**. The new **G** is constructed as

$$\mathbf{G}^* = (1 - \rho/2)\,\mathbf{G} + \mathbf{11}'\,\rho \qquad (37)$$

$\mathbf{G}^*$ is the adjusted genomic relationship matrix, **1** is a vector of ones, and ρ is Wright's F_{ST}, which models the difference between pedigree and genomic base.

When ssGBLUP was first implemented [28] in the BLUPF90 family of programs, $\mathbf{A}^{-1}$ was constructed based on Henderson [39] and Quaas [40] without considering inbreeding, $\mathbf{G}^{-1}$ was constructed based on VanRaden [7], and $\mathbf{A}_{22}^{-1}$ was based on Colleau [41]. As the algorithms to construct $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$ implicitly consider inbreeding, $\mathbf{H}^{-1}$ was often ill-conditioned because of the unbalance between $\mathbf{A}^{22}$ (i.e., the portion of $\mathbf{A}^{-1}$ for genotyped animals) and $\mathbf{A}_{22}^{-1}$, which has larger coefficients due to inbreeding. This would lead to convergence problems and overestimation of GEBV. To solve this problem, scaling factors to decrease the amount of information in $\mathbf{A}_{22}^{-1}$ (ω) and to increase in $\mathbf{G}^{-1}$ (τ) were proposed [28,42]:

$$\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tau\mathbf{G}^{-1} - \omega\mathbf{A}_{22}^{-1} \end{bmatrix} \qquad (38)$$

Primarily, ω controls inflation due to incompleteness of pedigree and τ controls additive genetic variance [43]. The ω parameter was usually set to 0.7 for beef and dairy cattle ssGBLUP

evaluations, and from 0.5 to 0.8 for pig evaluations. The appropriate value depended on the reduction of overestimation, which was evaluated based on validation studies. However, in 2016 the BLUPF90 developers decided to include inbreeding in $\mathbf{A}^{-1}$, and after that, the need for ω lower than 1 was reduced. In fact, when genotyped animals have complete pedigree, τ and ω are both equal to 1. Therefore, the compatibility among $\mathbf{A}^{-1}$, $\mathbf{G}^{-1}$, and $\mathbf{A}_{22}^{-1}$ is the key to avoid the use of ad-hoc scaling parameters.

2.6 Changing blending, tuning, and scaling parameters in `blupf90`

By default, in the `blupf90` the blending parameter α is set to 0.95, which makes 1-α (or β) equals to 0.05. This is used to overcome singularity problems (i.e., **G** being non-positive definite). Using lower values for α can speed up convergence, with small or no impact on accuracy. To change α and β in `blupf90`, assuming the new values would be 0.90 and 0.10, the following option can be added to renf90.par:

```
OPTION AlphaBeta 0.90 0.10
```

To model the difference between pedigree and genomic base, which is very important to reduce bias in GEBV, the default in the genomic library is to adjust **G** as proposed in Chen et al. [44]: $\mathbf{G}^{*} = \varphi\mathbf{G} + \delta$, where $\varphi = \left[\frac{\overline{diag\mathbf{A}_{22}} - \overline{offdiag\mathbf{A}_{22}}}{\overline{diag\mathbf{G}} - \overline{offdiag\mathbf{G}}}\right]$ and $\delta = \overline{diag\mathbf{A}_{22}} - \overline{diag\mathbf{G}} * \varphi$. To change the adjustment of **G** to the one proposed by Vitezica et al. [38] and demonstrated in equation (37), the following option can be added to the `blupf90` parameter file (e.g., renf90.par):

```
OPTION tunedG 4
```

A total of 4 different adjustments are implemented in the BLUPF90 family of programs; however types 2 [44] and 4 [38] are more frequently used. To see other options, check this link: http://nce.ads.uga.edu/wiki/doku.php?id=readme.pregsf90

If GEBV are underestimated/overestimated, ad-hoc scaling factors can be used to control the amount of information in $\mathbf{A}_{22}^{-1}$ (ω) and in $\mathbf{G}^{-1}$ (τ). The default in `blupf90` is ω= τ=1. To change those values, an option can be added to the `blupf90` parameter file. Supposing only ω is to be changed to 0.95, whereas τ is still 1:

```
OPTION TauOmega 1.0 0.95
```

Values of ω smaller than 1 helps to avoid overestimation; however, caution is recommended when using this option. A careful investigation of coefficients of the regression of a benchmark variable on GEBV in cross-validation studies is recommended when the objective is to choose an appropriate value.

2.7 Estimating SNP effects in ssGBLUP

Even though ssGBLUP is a genomic relationship-based method and provides GEBV as final output, SNP effects can still be calculated in this method. This is because GBLUP is equivalent to SNP-BLUP [7] as $\mathbf{u} = \mathbf{Za}$ and $Var(\mathbf{u}) = Var(\mathbf{Za})$. Using this idea, the selection index equation for GBLUP can be represented by:

$$\hat{\mathbf{u}} = \mathbf{G}\left[\mathbf{G} + \mathbf{R}\left(\frac{\sigma_a^2}{\sigma_e^2}\right)\right]^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) \qquad (39)$$

where **R** is a diagonal matrix accounting for heterogeneous residual variance. If $\hat{\mathbf{u}}|\hat{\mathbf{a}} = \mathbf{Z}\hat{\mathbf{a}}$, replacing the first **G** by $\mathbf{Z}'$, weighted by the ratio of SNP to additive direct variances (i.e., k = σ_a^2/σ_u^2), would allow the calculation of SNP effects (**a**) [7]:

$$\hat{\mathbf{a}} = \mathbf{Z}'\mathrm{k}\left[\mathbf{G} + \mathbf{R}\left(\frac{\sigma_a^2}{\sigma_e^2}\right)\right]^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) \qquad (40)$$

As we saw before, $\sigma_a^2 = \sigma_u^2/2\sum p_i(1-p_i)$. Therefore, k can be reduced to $1/2\sum p_i(1-p_i)$. Assuming that:

$$\mathbf{w} = \left[\mathbf{G} + \mathbf{R}\left(\frac{\sigma_u^2}{\sigma_e^2}\right)\right]^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) \qquad (41)$$

then,

$$\hat{\mathbf{a}} = \mathrm{k}\mathbf{Z}'\mathbf{w} \qquad (42)$$

and therefore,

$$\hat{\mathbf{u}} = \mathbf{G}\mathbf{w} \qquad (43)$$

In this way,

$$\mathbf{w} = \mathbf{G}^{-1}\hat{\mathbf{u}} \qquad (44)$$

Finally, the SNP effects can be calculated as:

$$\hat{\mathbf{a}} = \mathrm{k}\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}} \qquad (45)$$

as $Var(\mathbf{a}) = \mathbf{D}$, a diagonal matrix of SNP variance, the conditional mean of SNP effects given the GEBV is:

$$\hat{\mathbf{a}}|\hat{\mathbf{u}} = \mathrm{k}\mathbf{D}\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}} \qquad (46)$$

Thus, given GEBV from ssGBLUP are available, SNP effects are calculated as [45]:

$$\hat{\mathbf{a}} = \mathrm{k}\mathbf{D}\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}} \qquad (47)$$

In tests using simulated data sets the estimates of SNP effects were similar to those by BayesB [45]; however, the best estimates were not for a single SNP close to a QTL but for a cluster of nearby SNP.

If SNP effects are available, indirect predictions (IP) can be calculated for young genotyped animals in between official ssGBLUP evaluations, as the sum of SNP effects weighted by gene content [12]. Indirect predictions may also be useful for genotyped animals that have incomplete pedigree. Such animals can increase bias and reduce reliability of GEBV if included in official ssGBLUP evaluations, given their coefficients in **G** are not fully compatible to the ones in **A** [43]. Additionally, if lots of animals are genotyped but they do not contribute to the evaluation, having IP for them would reduce computing cost.

Another feature of having SNP effects is the ability to account for the fact SNP explain different proportion of genetic variance on the trait. By default, ssGBLUP (i.e., GBLUP-based methods) assume all SNP explain the same variance, so **D** is an identity matrix. However, Falconer, Mackay [46] showed the genetic variance explained by an additive locus (*i*) can be calculated as $2p_i(1-p_i)a_i^2$. Based on that, an iterative method was proposed to add different weights for SNP under ssGBLUP, which is called weighted ssGBLUP [45]. This method can be used to update only SNP effects or both SNP effects and GEBV. If both are updated, 7 steps are needed:

1. Set the diagonal matrix of SNP variance or weight as an identity, **D**=**I**

2. Compute the genomic relationships: **G**=**ZDZ**'/k, where $k = 1/2\sum p_i(1-p_i))$
3. Run ssGBLUP to obtain $\hat{\mathbf{u}}$
4. Convert $\hat{\mathbf{u}}$ into SNP effects: $\hat{\mathbf{a}} = k\mathbf{DZ}'\mathbf{G}^{-1}\hat{\mathbf{u}}$
5. Calculate SNP variance for SNP i as $d_i = 2p_i(1-p_i)a_i^2$ (i.e., quadratic weight)
6. Normalize **D**
7. Iterate from 2 until changes in SNP variance are small across iterations

Usually, the best weights are obtained after 1-2 rounds. Step 3 can be done once if changes in GEBV are small across iterations (i.e., updating only SNP effects). Different formulas can be used to calculate SNP variance (e.g., $d_i = a_i^2$). However, several authors have reported decrease in GEBV accuracy and increase in bias over iterations [47,48] when variance is calculated based on squared SNP effects, especially for more polygenic traits. This is because SNP variance would reach extreme values over iterations. VanRaden [7] proposed a formula to calculate SNP variance that limits the change over iterations, avoiding extreme values. This method is called non-linarA:

$$d_i = \mathrm{CT}^{\frac{|\hat{a}_i|}{\sigma(\hat{\mathbf{a}})}-2} \qquad (48)$$

where CT is a constant that determines the departure from normality; $|\hat{a}_i|$ is the absolute estimated SNP effect for marker i, and $\sigma(\hat{\mathbf{a}})$ is the standard deviation of the vector of estimated SNP effects. Garcia et al. [20] and Fragomeni et al. [49] showed that non-linearA had good convergence properties and avoided extreme values. The maximum change in variance is usually limited by the minimum between 5 and the exponent of CT; whereas CT was empirically derived as 1.125 over several polygenic traits for dairy cattle populations [7], meaning the distribution

for SNP effects approaches the normal distribution.

Considering SNP variance when constructing **G** in ssGBLUP seems to improve the accuracy of predicting GEBV for data sets with small number of genotyped animals, but marginal or no improvement was observed for large genotyped populations (i.e., >10k genotyped animals) [48], even for less polygenic traits. If the data allows to accurately estimate SNP effects, there is no advantage in selecting SNP and tagging chromosome segments differently. The fact that SNP selection does not improve accuracy with large datasets benefits commercial evaluations that use multiple-trait models, as models with different SNP per trait are easy to implement for single- but not multiple-trait models [50].

Once the variance for each SNP is calculated, the proportion of additive genetic variance can be plotted for all SNP in a Manhattan Plot. A threshold of 1% of genetic variance can be assumed if the objective is to explore associations between traits and regions in the genome, like in genome-wide association studies (GWAS). Additionally, p-values for SNP can be calculated as [51-53]:

$$pval_i = 2\left(1 - \Phi\left(\left|\frac{\hat{a}_i}{sd(\hat{a}_i)}\right|\right)\right) \quad (49)$$

where Φ is the cumulative standard normal function and $sd(\hat{a}_i)$ is the square root of prediction error variance (PEV) of the *i*-th SNP effect. Prediction error variance for each SNP effect can be calculated as [53]:

$$PEV(\hat{a}_i) = k\mathbf{z}_i'\mathbf{G}^{-1}\mathbf{z}_i\sigma_u^2 - \mathbf{z}_i'\mathbf{G}^{-1}\mathbf{C}^{u_2u_2}\mathbf{G}^{-1}\mathbf{z}_i \quad (50)$$

where $\mathbf{C}^{u_2u_2}$ is the portion of the inverse of the LHS of MME for ssGBLUP (34) referent to genotyped animals.

2.8 Using `postGSf90` to compute SNP effect, variance, and p-values

If the objective is to backsolve GEBV to SNP effect and then calculate variance explained by SNP, `postGSf90` can be used. This software was primarily developed to serve this purpose, but recently was modified to also compute p-values for SNP [54]. As this software relies on GEBV to calculate SNP effect and variance, `blupf90` needs to be run first with 2 additional options in renf90.par:

```
OPTION saveGInverse
OPTION snp_p_value
```

The first option saves $\mathbf{G}^{-1}$ and the second saves $\mathbf{C}^{u_2u_2}$, both in binary format. After running `blupf90`, renf90.par can be copied with another name, for example postgs.par, and the additional options are now:

```
OPTION readGInverse
OPTION snp_p_value
OPTION windows_variance 20
```

The first option is to read $\mathbf{G}^{-1}$, the second is use now to calculate p-values, and the third is used if variance for SNP is to be calculated based on windows of SNP instead of individual SNP [45]; windows of 20 SNP are generated in this example. Based on equation (47), `postGSf90` needs GEBV, SNP content, and $\mathbf{G}^{-1}$ to compute SNP effect and variance. The first one is obtained from the `blupf90` solutions file, the second from the SNP file, and the third from a file `blupf90` created and named Gi. For the calculation of p-values, a file containing $\mathbf{C}^{u_2u_2}$ was created by `blupf90` and named xx_ija. After running `postGSf90`, several files are generated. One is snp_sol, which the column information is described in Box 4.

Box 4: Content of snp_sol generated by `postGSf90`

1: Trait
2: Effect
3: SNP
4: Chromosome
5: Position
6: SNP effect
7: SNP variance
8: Variance explained by n adjacent SNP
(if OPTION windows_variance)
9: Variance of the SNP solution
(used to compute the p-value, if OPTION snp_p_value)

Three extra files are chrsnp, chrsnpvar, chrsnp_pval, which are used to generate Manhattan plots for SNP effect, proportion of variance explained by n adjacent SNP, and –log10(p-value), respectively. Additionally, R and gnuplot scripts are also generated to create the Manhattan plots described above. Box 5 shows how to generate Manhattan plots in R and gnuplot.

Box 5: Creating Manhattan plots from files generated by `postGSf90`

For R users:

```
Rscript Sft1e2.R # Creates Manhattan plots for SNP effect
Rscript Vft1e2.R # Creates Manhattan plots for SNP variance
Rscript Pft1e2.R # Creates Manhattan plots for SNP p-value
```

For gnuplot users:

```
gnuplot Sft1e2.gnuplot # Creates Manhattan plots for SNP effect
gnuplot Vft1e2.gnuplot # Creates Manhattan plots for SNP variance
gnuplot Pft1e2.gnuplot # Creates Manhattan plots for SNP p-value
```

The default formula to calculate variance or weight for SNP *i* is based on Falconer, Mackay [46], where $d_i = 2p_i(1-p_i)a_i^2$. However, 4 different formulas are implemented in

`postGSf90` (http://nce.ads.uga.edu/wiki/doku.php?id=readme.pregsf90). A more robust way to compute SNP variance is the non-linearA shown in equation (48). To change the SNP variance type to non-linearA, the following option should be added to the `postGSf90` parameter file:

```
OPTION which_weight nonlinearA
```

This option assumes the default constant (CT) is 1.125. To change the constant value to reflect a distribution closer to normal, use a CT value closer to 1:

```
OPTION which_weight nonlinearA 1.05
```

By default, the maximum change in SNP variance is limited to 5, which is calculated as $CT^{(5-2)}$ and returns a value of 1.4238 with CT=1.125. If this limit is to be changed to 10, the following option can be used, where the value provided (x) is the result of the expression $CT^{(x-2)}$. As an example, if CT is 1.05 and x is 10, the value provided to the option should be 1.4775:

```
OPTION SNP_variance_limit 1.4775
```

A parameter file to run `postGSf90` using non-linearA variance with CT equal to 1.05 and limit of 10, windows of 20 SNP, and computing p-value is in Box 6.

Box 6: Parameter file for running `postGSf90` using non-linearA variance, windows of 20 SNP, and computing p-value

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
      1
NUMBER_OF_EFFECTS
      2
OBSERVATION(S)
   1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS
TYPE_OF_EFFECT[EFFECT NESTED]
 2       2 cross
 3    12010 cross
RANDOM_RESIDUAL VALUES
  0.60000
 RANDOM_GROUP
    2
 RANDOM_TYPE
 add_an_upginb
 FILE
renadd02.ped
(CO)VARIANCES
  0.40000
OPTION SNP_file genotypes.txt
OPTION chrinfo gen_map.txt
OPTION readGInverse
OPTION snp_p_value
OPTION windows_variance 20
OPTION which_weight nonlinearA 1.05
OPTION SNP_variance_limit 1.4775
```

Although the calculation of SNP effect and variance was designed to be an iterative method, it is not recommended to use the iterative process when using the option to calculate p-value [54]. To check how to have weighted ssGBLUP where SNP effect, SNP variance, and GEBV are updated in an iterative way, see Note 4.

2.9 Accounting for sequence variants ssGBLUP

Genomic selection relies on linkage disequilibrium (LD) between SNP and quantitative trait nucleotide (QTN). By having dense SNP panels (i.e., >50,000 SNP), it is more likely that QTN will be in LD with at least one SNP. If QTN A is linked to SNP B, depending on the strength of this linkage, once SNP B is observed it will imply QTN A was inherited together. Therefore, it is expected that increasing the number of SNP the accuracy of genomic selection will increase. VanRaden et al. [55] showed an average increase of 1.6% in reliability of GEBV for a simulated trait when using 500,000 instead of 50,000 SNP. According to Meuwissen et al. [56], the ideal SNP density is given by whole-genome sequence data. As millions of SNP are screened, the causative variants are expected to be among them.

Using simulated data, Fragomeni et al. [57] showed that accuracy of GEBV in weighted ssGBLUP can approach 1 (i.e., perfect genomic prediction) if all causative variants are known and the true variance is assigned to each one of them. In a US Holstein dataset, Fragomeni et al. [49] tested the performance of ssGBLUP when using nearly 54,000 SNP and when adding 17,000 significant variants discovered in a GWAS that involved 33 traits [58]. Although VanRaden et al. [58] reported an increase in reliability of GEBV of 4.3 points for stature by using non-linearA weights in a multistep scenario, no gain was observed in Fragomeni et al. [49] using either quadratic or non-linearA weight in ssGBLUP. This is possibly because the amount of data used in ssGBLUP overwhelms any *a priori* assumption made about SNP effects, making this method less sensitive to SNP weighting in the presence of large data. Another hypothesis to explain the steady reliability is that not all causative variants were present among the 17,000 significant SNP. Although causative variants can be included in ssGBLUP assuming different

weights for SNP, maximizing the accuracy of GEBV would require the true identification of all causative variants and their substitution effect.

To check how to consider different weights or variance for causative variants in ssGBLUP, see Note 4.

2.10 Large-scale genomic evaluations with ssGBLUP

The most expensive operation in ssGBLUP, as implemented in Aguilar et al. [28] and Christensen, Lund [25], is the inversion of $\mathbf{G}$ and $\mathbf{A}_{22}$. This operation has an approximately cubic cost with the number of genotyped animals. With efficient computing algorithms, matrix inversion is feasible for up to 100,000 genotyped animals [59,60]. The number of genotyped animals in some livestock species goes far beyond 100,000 and considerably increases every year. One example is the American Angus Association that has over 600,000 (Steve Miller, 2019; personal communication) and the US dairy industry has already collected over 2.66M Holstein genotypes (https://queries.uscdcb.com/Genotype/counts.html), where only 11% of those are for males, over 75% are for animals without a BLUP evaluation, and there is a very slow increase in the number of genotypes for proven bulls [61].

To overcome the limitation set by the number of genotyped animals in ssGBLUP, Misztal et al. [62] proposed the algorithm for proven and young (APY) to construct $\mathbf{G}^{-1}$ without having to explicitly invert $\mathbf{G}$. The APY is based on the principles discovered by Henderson [39] and Quaas [40] to recursively construct the inverse of $\mathbf{A}$. The logic behind the construction of $\mathbf{G}_{\text{APY}}^{-1}$ is that the genotyped animals are split into core (*c*) and noncore (*n*), and the main assumption is

that breeding values for noncore animals ($\mathbf{u}_n$) are functions of breeding values of core animals ($\mathbf{u}_c$):

$$\mathbf{u}_n = \mathbf{P}_{nc}\mathbf{u}_c + \boldsymbol{\Psi}_n \quad (51)$$

where $\mathbf{P}_{nc}$ is a matrix that relates breeding values for noncore to core animals, and $\boldsymbol{\Psi}_n$ is a diagonal matrix with estimation errors. Following further developments [63], $\mathbf{G}^{-1}_{\text{APY}}$ can be constructed as:

$$\mathbf{G}^{-1}_{\text{APY}} = \begin{bmatrix} \mathbf{G}^{-1}_{cc} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{G}^{-1}_{cc}\mathbf{G}_{cn} \\ \text{I} \end{bmatrix} \mathbf{M}^{-1}_{nn} [-\mathbf{G}_{nc}\mathbf{G}^{-1}_{cc} \quad \text{I}] \quad (52)$$

with $m_{nn_{ii}} = g_{ii} - \mathbf{g}_{ic}\mathbf{G}^{-1}_{cc}\mathbf{g}_{ci}$. The APY algorithm creates a generalized sparse inverse of **G** at approximately a linear cost in computing and storage [63,62] and has been extensively tested for beef cattle [12], dairy cattle [64,65], and pigs [66,67]. This algorithm enables ssGBLUP evaluations with millions of genotyped animals, as the only inverse needed is for the core animals. Pocrnic et al. [68] and Pocrnic et al. [69] found that the ideal number of core animals depends on the dimensionality of genomic information. Even though millions of animals can be genotyped, the amount independent genomic information or independent chromosome segments is limited and depends on the effective population size (Ne) and genome length. The knowledge about this non-redundant information enables computations with large-scale genomic data. Pocrnic et al. [69] found that the optimal number of core animals around 4000 for pigs and chicken, 11,000 for Angus, 12,000 for Jerseys, and 14,000 for Holsteins.

If $\mathbf{G}^{-1}_{\text{APY}}$ is efficiently computed but $\mathbf{A}^{-1}_{22}$ is not, ssGBLUP cannot be used for over 100,000 genotyped animals. To avoid explicit inversion of $\mathbf{A}_{22}$, Stranden, Mantysaari [70] and Masuda et al. [71] proposed to compute an efficient inverse indirectly as a product of sparse matrices:

$$\mathbf{A}_{22}^{-1} = \mathbf{A}^{22} - \mathbf{A}^{21}(\mathbf{A}^{11})^{-1}\mathbf{A}^{12} \qquad (53)$$

where $\mathbf{A}^{11}$, $\mathbf{A}^{21}$, and $\mathbf{A}^{22}$ are portions of $\mathbf{A}^{-1}$ for non-genotyped, between genotyped and non-genotyped, and for genotyped animals, respectively. Computing time for constructing $\mathbf{A}_{22}^{-1}$ for 570,000 genotyped animals extracted from a population of 10M animals was around 11 minutes [71]. Single-step GBLUP with $\mathbf{G}_{\text{APY}}^{-1}$ and efficient $\mathbf{A}_{22}^{-1}$ was successfully applied to over 21M cows with milking records, 30M animals in the pedigree, and about 764,000 genotyped Holsteins [72]; using a little over 18,000 core animals, the complete evaluation for a 3-trait model that included milk, fat, and protein yields took less than 24 hours.

Unfortunately, the subroutine to create $\mathbf{G}_{\text{APY}}^{-1}$ is not implemented in the free distribution of BLUPF90 family of programs.

2.11 UPG and metafounders in ssGBLUP

Commercial populations especially beef and dairy cattle often have incomplete pedigrees. In BLUP, missing parents are modeled by UPG [40,73,74]. Such groups are also known as phantom parents or genetic groups, and are used to represent the average level of breeding value in a group where parents were missing. Different groups can be assigned based on year of birth, sex, breed combination, etc. As UPG are mainly modeled as fixed effects, they need to be defined carefully to avoid confounding with other effects in the model [40]. In ssGBLUP, when UPG are applied only to pedigree relationship, the convergence rate can be slow or no convergence may be reached [75]. Misztal et al. [76] revised UPG equations to include groups also in the genomic portion of $\mathbf{H}^{-1}$, which then becomes $\mathbf{H}^{-1*}$, based on Quaas-Pollak (QP) transformation [73]:

$$\mathbf{H}^{-1*} = \mathbf{A}^{-1*} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}_{22}^{-1} & -(\mathbf{G}^{-1} - \mathbf{A}_{22}^{-1})\mathbf{Q}_2 \\ \mathbf{0} & -\mathbf{Q}_2'(\mathbf{G}^{-1} - \mathbf{A}_{22}^{-1}) & \mathbf{Q}_2'(\mathbf{G}^{-1} - \mathbf{A}_{22}^{-1})\mathbf{Q}_2 \end{bmatrix} \quad (54)$$

where $\mathbf{Q}_2$ is a matrix that relates genotyped animals to groups; $\mathbf{G}^{-1}$ can be replaced by $\mathbf{G}_{APY}^{-1}$ in large genotyped populations. When UPGs were applied to all components of $\mathbf{H}^{-1}$, convergence dramatically improved for a multiple-trait model in the Nordic dairy cattle population [77]. Revised UPGs also worked well for the US Holstein data up to 2014 [43]. However, using data updated to 2015, Masuda et al. [78], reported lower prediction reliabilities using revised UPG than not using UPG at all. Therefore, it is not clear whether ssGBLUP equations should include UPG for **G**, as genomic relationships are not affected by missing pedigree, implying UPG are automatically accounted for.

Current use of UPG in BLUP ignores the fact they represent sets of related, missing parents in a population under constant selection. Thus, a more accurate modelling would assume missing parents can be related and inbred [79,80]. Legarra et al. [81] proposed the idea of metafounders, which are "inbred and related" UPG. In ssGBLUP, the genomic relationships are usually derived based on current allele frequencies and scaled for compatibility with pedigree relationships as in Vitezica et al. [38]. Based on the metafounders theory, **G** would be derived using 0.5 allele frequencies as an "absolute reference" [82], and **A** would be scaled for compatibility with **G** using covariances among and within metafounders. According to Legarra et al. [81] the covariances represent size of the base population at the time when pedigree recording started and they would be estimated in such a way so that they account for scaling, unaccounted inbreeding, and different genetic level (i.e., when using multibreed or selected populations). Several methods were proposed to estimate the covariances among metafounders, including via gene frequencies related to

unknown parents [83]. In simulations and real data, the concept of metafounders delivered the least biased predictions [83,84]. In ssGBLUP, $\mathbf{H}^{-1}$ with metafounders ($\mathbf{H}^{\Gamma-1}$) can be represented by:

$$\mathbf{H}^{\Gamma-1}=\mathbf{A}^{\Gamma-1}+\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{G}^{-1}-\mathbf{A}_{22}^{\Gamma-1}\end{bmatrix} \qquad (55)$$

where Γ is the relationship matrix among metafounders, which can be estimated using generalized least squares [83]. Once Γ is inverted, Henderson [39] rules can be used to construct the inverse of the pedigree relationship matrix.

In the BLUPF90 family of programs, `renumf90` can create UPG based on year of birth or can recognize negative values in the pedigree as UPG (see Note 2). If `blupf90` is used to run ssGBLUP, UPG will be set only for $\mathbf{A}^{-1}$. To set UPG for the full $\mathbf{H}^{-1}$like demonstrated in (54), the following extra option is needed:

```
OPTION exact_upg
```

As the metafounders concept is still recent, the BLUPF90 developers are currently working on a standalone software to estimate Γ. After that, `blupf90` will be modified to accept an extra type of random effect specific for metafounders. Independent software used in Garcia-Baccino et al. [83] to estimate Γ and to compute $\mathbf{H}^{\Gamma-1}$ with instructions and examples can be found here https://github.com/alegarra/metafounders. After $\mathbf{H}^{\Gamma-1}$ is constructed, `blupf90` can be used with the random type set as `user_file` (see `RANDOM_TYPE` in Note 2). This allows `blupf90` to use external covariance matrices half-stored as row, column, and value (in Ascii, i.e., plain text). For more details on how to use `user_file`, check the following link: http://nce.ads.uga.edu/wiki/doku.php?id=user_defined_files_for_covariances_of_random_effects

3 Notes

1. Downloading and executing programs from the BLUPF90 software suite

a) Follow the link to the official web site of the Animal Breeding and Genetics Group at the University of Georgia to access the binaries: http://nce.ads.uga.edu/html/projects/programs/;

b) Select the desired operation system;

c) Download the desired program and store it in folder. Add this folder to a PATH or copy the programs to the same folder where the data files for analysis are stored;

d) Open a Terminal or Command Prompt window;

e) Type the name of the program to run it (i.e., `blupf90` for Linux and Mac or `blupf90.exe` for Windows);

f) The program will ask for the name of the parameter file. Type the name of the parameter file and hit ENTER key;

g) Wait for the program to finish and check the output in the screen.

If there is a need to save the screen output to a file, type:

echo parameter_file.par | `blupf90` | tee out.log

The above command will provide the parameter file when `blupf90` asks for it and will save the screen output to a file named out.log

2. Renumbering the data with `renumf90`:

BLUPF90 software suite only works with numeric entries (i.e., integer or real) and levels of all effects need to be consecutive starting form 1. In field datasets, animal ID contains alphanumeric characters and levels of fixed effects are combinations of two or more effects (i.e., contemporary groups). To avoid extra work with sorting and renumbering all effects using independent scripts, `renumf90` can be used to renumber the data. This software creates a renumbered phenotype (renf90.dat) and pedigree files (renaddXX.ped; where XX refers to the number of the animal effect in the model), along with a cross-reference table for fixed effects (renf90.tables), a cross-reference file for IDs of genotyped animals (*name_of_snp_file*_XrefID), and a file with inbreeding coefficients if inbreeding is used to compute $\mathbf{A}^{-1}$. One interesting feature of `renumf90` is that it can trace pedigree back *n* generations for animals in the data and/or SNP file.

The `renumf90` requires a parameter file that consists of keywords (capital letters) and the corresponding values. There are 6 mandatory keywords: DATAFILE, TRAITS, FIELDS_PASSED TO OUTPUT, WEIGHT(S), RESIDUAL_VARIANCE, and EFFECT. If there is no need to use FIELDS_PASSED TO OUTPUT and WEIGHT(S), simply put an empty line as a value.

Keyword	Possible Value	Description
`DATAFILE`	characters	Name of the data file to be used (should be space-delimited file)
`TRAITS`	integer	Position of traits in the data file
`FIELDS_PASSED TO OUTPUT`	integer	Columns to pass to the new data file without renumbering

WEIGHT(S)	integer	Position of weight column in the data file. Weights for the residual variance
RESIDUAL_VARIANCE	real	Residual (co)variances
EFFECT	integer	Description of the effects in the model. Each effect should be described with a keyword EFFECT

The EFFECT keyword has several values that are described below:

Keyword	Position	Type	Data Type
EFFECT	integer	cross cov	alpha or number

Where position means the column number for the effect being described; effect type is *cross* for cross-classified effect and *cov* for covariables. If cross-classified, a data type alpha or numer is required to describe variables created based on alphanumeric or only numeric characters, respectively.

Optional keywords also exist and if used, should follow a specific order. For example, if an effect is random, the keyword RANDOM and its value should follow the EFFECT description. Here are the possible optional keywords. A full description can be found in the BLUPF90 manual (http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=blupf90_all7.pdf):

Keyword	Description/possible values
NESTED	Covariables can be nested in cross-classified effects
RANDOM	Declaration of random effects; can be diagonal (non-correlated) or animal (correlated)
OPTIONAL	Used to create permanent environmental (PE), maternal (MAT) and maternal permanent environmental (MPE)

FILE	Name of the raw pedigree file (for RANDOM animal)
FILE_POS	Positions of animal, sire, dam, surrogate dam, year of birth in the pedigree file
SNP_FILE	Name of SNP marker file (if genomic information is available)
PED_DEPTH	Number of generations to trace the pedigree back for animals with phenotypes and/or genotypes. If 0, all animals are passed to the new pedigree file. If no input, the default value is 3
GEN_INT	Minimum, average, and maximum generation interval (GI). GI is used to set unknown parent groups (UPG)
REC_SEX	If only one sex has records, specifies which parent it is; used for pedigree checks
UPG_TYPE	'yob' = based on year of birth 'in_pedigrees' = the value of a missing parent should be -x, where x is UPG number that this missing parent should be allocated to
INBREEDING	To consider inbreeding for A^{-1} 'pedigree' = calculated from pedigree 'file_with_inb.txt' to provide a file with two columns: animal ID and inbreeding coefficient
RANDOM_REGRESSION	Put covariates for random regressions 'data' = covariates are in the data
RR_POSITION	Position of covariates in the data
(CO)VARIANCES	(Co)variance components for general random effects
(CO)VARIANCES_PE	(Co)variance components for permanent environmental effect
(CO)VARIANCES_MPE	(Co)variance components for maternal permanent environmental effect
OPTION	Any extra option that the BLUPF90 family of programs can take. To see other options, check the online manual

The following parameter file can be used to use `renumf90` to renumber the data described in section 2.4, following the model y = sex + u + residual, which considers sex as fixed

and u (i.e., animal effect or direct additive genetic effect) as random (remember that phenotypes.txt contains 5 columns: animal, sex, phenotype, true breeding value, generation):

```
DATAFILE
phenotypes.txt
TRAITS
 3
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
0.60
EFFECT
2 cross alpha
EFFECT
1  cross alpha
RANDOM
animal
FILE
pedigree.txt
FILE_POS
1 2 3 0 0
SNP_FILE
genotypes.txt
PED_DEPTH
0
INBREEDING
pedigree
(CO)VARIANCES
0.40
OPTION chrinfo gen_map.txt
```

Hint 1: Usually fixed effects are declared before random effects.

Hint 2: Do not leave blank spaces between keywords and values or vice-versa. Blank spaces are only allowed below `FIELDS_PASSED TO OUTPUT` and `WEIGHT(S)` if there is no input. The program will stop if other blank spaces are detected.

Hint 3: Save the parameter file above (e.g., parameter1.par). To run `renumf90` and save the screen output to a file, use the following command line:

`echo` parameter1.par | `renumf90` | `tee` out.log

Hint 4: `OPTION chrinfo` is not mandatory. It is used to provide the program the name of the SNP map file. This option will be passed by `renumf90` to the new parameter file without being used.

The renumbered files and contents are:

1) renf90.dat – is the renumbered phenotype file and contains 3 columns: phenotype, renumbered sex code, renumbered animal ID;
2) renadd02.ped – is the renumbered pedigree file and contains 10 columns:
   i) renumbered animal ID (from 1);
   ii) renumbered sire ID (of parent 1 ID);
   iii) renumbered dam ID (or parent 2 ID);
   iv) 3 minus number of known parents (or inbreeding code if keyword INBREEDING is used);
   v) known or estimated year of birth (0 if not provided);
   vi) number of known parents (if animal has genotype, it is 10+number of know parents);
   vii) number of records;
   viii) number of progeny as parent 1;
   ix) number of progeny as parent 2;
   x) original animal id.
3) renf90.tables – is a file with correspondence table between the original code for fixed effects and the renumbered value. It is organized in 3 columns: code, number of observations, renumbered value.
4) renf90.inb – contains the animal original ID and the inbreeding coefficient.
5) genotypes.txt_XrefID – is a cross-reference file with renumbered ID and original ID. This file is created to avoid editing the SNP file, which is usually big and requires a lot of memory. By default, the name of this file is a concatenation of the name of SNP file and the suffix “XrefID”, which means cross-reference ID.

6) renf90.par – is the new parameter file that can be used for all other programs from the BLUPF90 family. This is how renf90.par looks like for the simulated data:

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
      1
NUMBER_OF_EFFECTS
      2
OBSERVATION(S)
   1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS
TYPE_OF_EFFECT[EFFECT NESTED]
 2      2 cross
 3    12010 cross
RANDOM_RESIDUAL VALUES
  0.60000
 RANDOM_GROUP
    2
 RANDOM_TYPE
 add_an_upginb
 FILE
renadd02.ped
(CO)VARIANCES
 0.40000
OPTION SNP_file genotypes.txt
OPTION chrinfo gen_map.txt
```

The parameter file generated by `renumf90` is also based on keywords and values that are described below:

Keyword	Description/possible values
`DATAFILE`	Name of the file with phenotypes (space-delimited file)
`NUMBER_OF_TRAITS`	Number of traits to be analyzed
`NUMBER_OF_EFFECTS`	Number of effects in the model (does not account for the residual effect)
`OBSERVATION(S)`	Column number for the phenotype(s) in the data file

WEIGHT(S)	Column number for weights in the data file (leave a blank space if no weight)
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT NESTED]	Description of each effect in the model. Includes: column number for the effect in the data file, number of levels for the effect, type of effect (cross or cov). If a covariable effect is nested, the column number of the effect in which the covariable is nested will be displayed.
RANDOM_RESIDUAL_VALUE	Residual variance (or covariance if 2 or more traits)
RANDOM_GROUP	Sequential effect number for a random effect (the order that the effect is shown in the EFFECTS section)
RANDOM_TYPE	Type of random effect: diagonal, add_sire, add_an_upg, add_an_upginb, par_domin, or user_file. If inbreeding is used, RANDOM_TYPE is add_an_upginb.
FILE	Pedigree file or other file associated with the random effect; blank if no file or if RANDOM_TYPE is diagonal
(CO)VARIANCES	Variance for the random effect (or covariance if 2 or more traits; a covariance matrix is also required when additive genetic direct and maternal are used)
OPTION SNP_file	Need to be followed by the name of the SNP marker file. This option is used to run ssGBLUP. Without it, genomic information is not used.
OPTION chrinfo	Need to be followed by the name of the SNP map file.
OPTION	Any extra option that the BLUPF90 family of programs can take. To see other options, check the online manual

3. Quality control of genomic data with `preGSf90`:

This software is an interface for the genomic library and contains several options (http://nce.ads.uga.edu/wiki/doku.php?id=readme.pregsf90). One useful task is to perform a quality control of the genomic data, removing SNP with low minor allele frequency (MAF) and monomorphic, SNP departing from the Hardy-Weinberg Equilibrium, and SNP with low call rate (i.e., missing in several samples). Checks for animals are also done, which includes removing

animals with low call rate (i.e., several SNP are missing) and animals with Mendelian conflicts (i.e., parentage verification). One option allows `preGSf90` to save the clean SNP and SNP map files: `OPTION saveCleanSNPs`. As `preGSf90` also constructs **G** and $\mathbf{A}_{22}$, respective inverses, and $\mathbf{G}^{-1}$- $\mathbf{A}_{22}^{-1}$, some extra options can be used to force the program to perform only quality control, avoiding the creation of those matrices. The options are `OPTION createG 0`, `OPTION createA22 0`, `OPTION createGInverse 0`, `OPTION createA22Inverse 0`, `OPTION createGimA22i 0`. Therefore, the parameter file to perform only quality control in preGSf90 can be constructed by adding extra options to renf90.par.

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
      1
NUMBER_OF_EFFECTS
      2
OBSERVATION(S)
   1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS
TYPE_OF_EFFECT[EFFECT NESTED]
 2       2 cross
 3    12010 cross
RANDOM_RESIDUAL VALUES
  0.60000
 RANDOM_GROUP
    2
 RANDOM_TYPE
 add_an_upginb
 FILE
renadd02.ped
(CO)VARIANCES
  0.40000
OPTION SNP_file genotypes.txt
OPTION chrinfo gen_map.txt
OPTION saveCleanSNPs
```

```
OPTION createG 0
OPTION createA22 0
OPTION createGInverse 0
OPTION createA22Inverse 0
OPTION createGimA22i 0
```

To run `preGSf90` and save the screen output to a file, use the following command line:

`echo` renf90.par | `preGSf90` | `tee` preGSout.log

If a SNP map file is provided, 3 new files are generated by `preGSf90`: genotypes.txt_clean, geotypes.txt_XrefID_clean, and gen_map.txt_clean, which contain the information after removing SNP and animals that did not pass the quality control. If `preGSf90` is used to do the quality control and save clean files, the parameter file for the subsequent run of `blupf90` should include the name of the clean file and an option to avoid running the quality control again:

```
OPTION SNP_file genotypes.txt_clean
OPTION chrinfo gen_map.txt_clean
OPTION no_quality_control
```

The `preGSf90` software can also be used to calculate linkage disequilibrium, to do single value decomposition of the SNP file, to plot the first two principal components for population structure checks, to calculate heritability of gene content, and to save relationship matrices in text or binary formats, including $\mathbf{H}^{-1}$. As **H** is not needed in ssGBLUP, this matrix cannot be created using `preGSf90`.

4. Iterative weighted ssGBLUP with `blupf90` and `postGSf90`:

Variance or weights for SNP can be used to construct the genomic relationship matrix when a diagonal matrix of weights is included in the equation to create $\mathbf{G}$ [7], as $\mathbf{G} = \frac{\mathbf{ZDZ'}}{2\sum p_i(1-p_i)}$. If $\mathbf{G}$ can be updated with weights, the fact SNP explain different proportion of variance can be extended to GEBV. If weights are proper, GEBV accuracy may increase, but this increase depends on data structure. Large genomic data seems to do not benefit from different SNP weighting [48].

To iteratively calculate and use weights to update GEBV and SNP effect/variance, it is recommended to use the non-linearA option to avoid extreme SNP variance [49,20]. Assuming the data is renumbered (see Note 2) and updates are done for GEBV and SNP effect/variance, `blupf90` and `postGSf90` should be run consecutively until changes in SNP variance or GEBV between the previous and current iteration are small [49]. The parameter file for `blupf90` should include one extra option to save $\mathbf{G}^{-1}$ (`OPTION saveGInverse`), to use weights for SNP (`OPTION weightedG` weights.txt), and to avoid quality control (`OPTION no_quality_control`). Avoiding quality control in the iterative run is important, so the number of SNP will be the same in consecutive iterations. The file weights.txt contains a column with weights for SNP, where the number of lines is the number of SNP. In the first iteration, this file contains a column of 1's. The parameter file for `blupf90` becomes:

```
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
      1
NUMBER_OF_EFFECTS
      2
OBSERVATION(S)
   1
WEIGHT(S)
```

```
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS
TYPE_OF_EFFECT[EFFECT NESTED]
 2      2 cross
 3    12010 cross
RANDOM_RESIDUAL VALUES
 0.60000
RANDOM_GROUP
   2
RANDOM_TYPE
add_an_upginb
FILE
renadd02.ped
(CO)VARIANCES
 0.40000
OPTION SNP_file genotypes.txt
OPTION chrinfo gen_map.txt
OPTION saveGInverse
OPTION weightedG weights.txt
OPTION no_quality_control
```

After running `blupf90`, run `postGSf90` with the following parameter file, which assumes default values for the constant and limit in non-linearA, and a windows variance of 20 SNP:

```
DATAFILE
renf90.dat
NUMBER_OF_TRAITS
     1
NUMBER_OF_EFFECTS
     2
OBSERVATION(S)
  1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS
TYPE_OF_EFFECT[EFFECT NESTED]
 2      2 cross
 3    12010 cross
RANDOM_RESIDUAL VALUES
 0.60000
RANDOM_GROUP
   2
RANDOM_TYPE
add_an_upginb
FILE
```

```
renadd02.ped
(CO)VARIANCES
 0.40000
OPTION SNP_file genotypes.txt
OPTION chrinfo gen_map.txt
OPTION readGInverse
OPTION windows_variance 20
OPTION which_weight nonlinearA
OPTION weightedG weights.txt
OPTION no_quality_control
```

After running `postGSf90`, the new SNP weight or variance will be the column number 7 of snp_sol. Save this column as weights.txt and start a new iteration of `blupf90` and `postGSf90` until changes in SNP variance or in GEBV are small.

Sometimes during the iterations, `blupf90` outputs a warning "correlation for off-diagonals **G** and $\mathbf{A}_{22}$ is lower than 0.5", especially when the default formula to calculate weights is used (i.e., $d_i = 2p_i(1-p_i)a_i^2$). This is because using weights for **G** can create some extreme values, lowering the correlation with $\mathbf{A}_{22}$. This correlation is expected to be from 0.5 to 0.9, where values greater than 0.9 indicate information in **G** and $\mathbf{A}_{22}$ is very similar, therefore, a small gain in accuracy is expected by using genomic information. Low correlation in the first iteration may be a sign of misidentified or low quality genomic samples.

4. References

1. Soller M, Beckmann JS (1983) Genetic polymorphism in varietal identification and genetic improvement. Theor Appl Genet 67:25-33
2. The International SNP Map Working Group (2001) A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. Nature 409:928-933
3. Stonecking M (2001) From the evolutionary past... . Nature 409:821-822
4. Schork NJ, D.Fallin, Lanchbury S (2000) Single nucleotide polymorphisms and the future of genetic epidemiology. Clin Genet 58:250-264
5. Meuwissen THE, Hayes BJ, Goddard ME (2001) Prediction of total genetic value using genome-wide dense marker maps. Genetics 157:1819-1829
6. Fernando RL, Grossman M (1989) Marker-assisted selection using best linear unbiased prediction. Genet Sel Evol 21:467-477

7. VanRaden PM (2008) Efficient methods to compute genomic predictions. Journal of Dairy Science 91:4414-4423. doi:10.3168/jds.2007-0980
8. Wiggans GR, Cooper TA, VanRaden PM, Cole JB (2011) Technical note: Adjustment of traditional cow evaluations to improve accuracy of genomic predictions. J Dairy Sci 94:6188-6193
9. Wiggans GR, VanRaden PM, Cooper TA (2012) Technical note: Adjustment of all cow evaluations for yield traits to be comparable with bull evaluations. J Dairy Sci 95:3444-3447
10. Misztal I, Legarra A, Aguilar I (2009) Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information. Journal of Dairy Science 92:4648-4655. doi:10.3168/jds.2009-2064
11. Legarra A, Aguilar I, Misztal I (2009) A relationship matrix including full pedigree and genomic information. J Dairy Sci 92:4656-4663
12. Lourenco DAL, Tsuruta S, Fragomeni BO, Masuda Y, Aguilar I, Legarra A, Bertrand JK, Amen T, Wang L, Moser DW, Misztal I (2015) Genetic evaluation using single-step genomic best linear unbiased predictor in American Angus. Journal of Animal Science 93:2653-2662. doi:10.2527/jas.2014-8836
13. Forni S, Aguilar I, Misztal I (2011) Different genomic relationship matrices for single-step analysis using phenotypic, pedigree, and genomic information. Genet Sel Evol 43:1
14. Lourenco DAL, Tsuruta S, Fragomeni BO, Chen CY, Herring WO, Misztal I (2016) Crossbreed evaluations in single-step genomic best linear unbiased predictor using adjusted realized relationship matrices. Journal of Animal Science 94:909-919. doi:10.2527/jas.2015-9748
15. Chen CY, Misztal I, Aguilar I, Tsuruta S, Meuwissen THE, Aggrey SE, Wing T, Muir WM (2011) Genome-wide marker-assisted selection combining all pedigree phenotypic information with genotypic data in one step: an example using broiler chickens. J Anim Sci 89:23-28
16. Lourenco DAL, Fragomeni BO, Tsuruta S, Aguilar I, Zumbach B, Hawken RJ, Legarra A, Misztal I (2015) Accuracy of estimated breeding values with genomic information on males, females, or both: an example in broiler chicken. Genet Sel Evol 47:56
17. Yan Y, Wu G, Liu A, Sun C, Han W, Li G, Yang N (2018) Genomic prediction in a nuclear population of layers using single-step models. Poult Sci 97:397-402
18. Rupp R, Mucha S, Larroque H, McEwan J, Conington J (2016) Genomic application in sheep and goat breeding. Anim Frontiers 6:39-44
19. Brown DJ, Swan AA, Boerner V, Li L, Gurman PM, McMillan AJ, van der Werf JHJ, Chandler HR, B. Tier, Banks RG (2018) Single-Step Genetic Evaluations in the Australian Sheep Industry. In Proc 11th World Congress on Gen Appl to Livest Prod Auckland, New Zealand, 11-16 February 2018
20. Garcia ALS, Bosworth B, Waldbieser G, Misztal I, Tsuruta S, Lourenco DAL (2018) Development of genomic predictions for harvest weight and carcass weight in channel catfish. Genet Sel Evol 50:66
21. Aguilar I, Tsuruta S, Masuda Y, Lourenco DAL, Legarra A, Misztal I (2018) BLUPF90 suite of programs for animal breeding with focus on genomics. In Proc 11th World Congress on Gen Appl to Livest Prod Auckland, New Zealand, 11-16 February 2018
22. Legarra A, Lourenco DAL, Vitezica Z (2018) Bases for genomic predictions. http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=gsip.pdf.
23. Leutenegger A-L, Prum B, Génin E, Verny C, Lemainque A, Clerget-Darpoux F, Thompson EA (2003) Estimation of the inbreeding coefficient through use of genomic data. Am J Hum Genet 73:516-523
24. Amin N, Duijn CMv, Aulchenko YS (2007) A genomic background based method for association analysis in related individuals. PLoS ONE 2:e1274
25. Christensen OF, Lund MS (2010) Genomic prediction when some animals are not genotyped. Genet Sel Evol 42:2
26. Legarra A, Chistensen OF, Aguilar I, Misztal I (2014) Single step, a general approach for genomic selection. Livest Prod Sci 166:54-65

27. Gengler N, Mayeres P, Szydlowski M (2007) A simple method to approximate gene content in large pedigree populations: application to the myostatin gene in dual-purpose Belgian Blue cattle. Animal 1:21-28
28. Aguilar I, Misztal I, Johnson DL, Legarra A, Tsuruta S, Lawlor TJ (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. doi:10.3168/jds.2009-2730
29. Patterson HD, Thompson R (1971) Recovery of inter-block information when block sizes are unequal. Biometrika 58:545-554
30. Baloche G, Legarra A, Salle G, Larroque H, Astruc JM, Robert-Granie C, Barillet F (2014) Assessment of accuracy of genomic prediction for French Lacaune dairy sheep. J Dairy Sci 97:1107–1116
31. Christensen OF, Madsen P, Nielsen B, Ostersen T, Su G (2012) Single-step methods for genomic evaluation in pigs. Animal 6:1565-1571
32. Gray KA, Cassady JP, Huang Y, Maltecca C (2012) Effectiveness of genomic prediction on milk flow traits in dairy cattle. Genet Sel Evol 44:24
33. Lourenco DAL, Misztal I, Tsuruta S, Aguilar I, Ezra E, Ron M, Shirak A, Weller J (2014) Methods for genomic evaluation of a relatively small genotyped dairy population and effect of genotyped cow information in multiparity analyses. Journal of Dairy Science 97:1742-1752
34. Tsuruta S, Misztal I, Lawlor TJ (2013) Short communication: Genomic evaluations of final score for US Holsteins benefit from the inclusion of genotypes on cows. J Dairy Sci 96:3332-3335
35. Patry C, Ducrocq V (2011) Evidence of biases in genetic evaluations due to genomic preselection in dairy cattle. J Dairy Sci 94:1011-1020
36. Sargolzaei M, Schenkel FS (2009) QMSim: A large-scale genome simulator for livestock. Bioinformatics 25:680-681
37. Tsuruta S, Misztal I, stranden I (2001) Use of the preconditioned conjugate gradient algorithm as a generic solver for mixed-model equations in animal breeding applications. J Anim Sci 79:1166-1172
38. Vitezica ZG, Aguilar I, Misztal I, Legarra A (2011) Bias in genomic predictions for populations under selection. Genet Res (Camb) 93:357-366
39. Henderson CR (1976) A simple method for computing the inverse of a relationship matrix used in prediction of breeding values. Biometrics 32:69
40. Quaas RL (1988) Additive genetic model with groups and relationships. J Dairy Sci 71:1338-1345
41. Colleau JJ (2002) An indirect approach to the extensive calculation of relationship coefcients. Genet Sel Evol 34:409-421
42. Tsuruta S, Misztal I, Aguilar I, Lawlor TJ (2011) Multiple-trait genomic evaluation of linear type traits using genomic and phenotypic data in US Holsteins. J Dairy Sci 94:4198-4204
43. Misztal I, Bradford HL, Lourenco DAL, Tsuruta S, Masuda Y, Legarra A, Lawlor TJ (2017) Studies on Inflation of GEBV in Single-Step GBLUP for Type. Interbull Bull 51:38-42
44. Chen CY, Misztal I, Aguilar I, Legarra A, Muir WM (2011) Effect of different genomic relationship matrices on accuracy and scale. J Anim Sci 89:2673-2679
45. Wang H, Misztal I, Aguilar I, Legarra A, Muir WM (2012) Genome-wide association mapping including phenotypes from relatives without genotypes. Genetics Research 94:73-83. doi:10.1017/S0016672312000274
46. Falconer D, Mackay T (1996) Introduction to quantitative genetics. 4th edn. Longman Green, Harlow, Essex, UK
47. Lee J, Cheng H, Garrick D, Golden B, Dekkers J, Park K, Lee D, Fernando R (2017) Comparison of alternative approaches to single-trait genomic prediction using genotyped and non-genotyped Hanwoo beef cattle. Genet Sel Evol 49:2

48. Lourenco DAL, Fragomeni BO, Bradford HL, Menezes IR, Ferraz JBS, Tsuruta S, Aguilar I (2017) Implications of SNP weighting on single-step genomic predictions for different reference population sizes. J Anim Breed Genet 134:463-471
49. Fragomeni BO, Lourenco DAL, Legarra A, VanRaden PM, Misztal I (2019) Alternative SNP weighting for SSGBLUP evaluation of stature in US Holstein in the presence of selected sequence variants. J Dairy Sci (under review)
50. Tiezzi F, Maltecca C (2015) Accounting for trait architecture in genomic predictions of US Holstein cattle using a weighted realized relationship matrix. Genet Sel Evol 47:24
51. Aguilar I, Legarra A, Cardoso F, Masuda Y, Lourenco D, Misztal I (2019) Frequentist p-values for large-scale single step genome-wide association, with an application to birth weight in American Angus. Genet Sel Evol (Under review)
52. Bernal-Rubio YL, Gualdron-Duarte JL, Bates RO, Ernst CW, Nonneman D, Rohrer GA, King A, Shackelford SD, Wheeler TL, Cantet RJC, Steibel JP (2015) Meta-analysis of genome-wide association from genomic prediction models. Anim Genet 47:36-48
53. Gualdron-Duarte JL, Cantet RJC, Bates RO, Ernest CW, Raney NE, Steibel JP (2014) Rapid screening for phenotype-genotype associations by linear transformations of genomic evaluations. BMC Bioinformatics 15:246
54. Aguilar I, Legarra A, Cardoso F, Masuda Y, Lourenco D, Misztal I (2019) Frequentist p-values for large-scale single step genome-wide association, with an application to birth weight in American Angus. Genet Sel Evol Under review
55. VanRaden PM, O'Connell JR, Wiggans GR, Weigel KA (2011) Genomic evaluations with many more genotypes. Genetic Selection Evolution 43:10
56. Meuwissen THE, Hayes B, Goddard M (2016) Genomic selection: A paradigm shift in animal breeding. Anim Front 6:6-14
57. Fragomeni BO, Lourenco DAL, Masuda Y, Misztal I (2017) Incorporation of Causative Quantitative Trait Nucleotides in Single-step GBLUP. Genetic Selection Evolution *(In Press)*
58. VanRaden PM, Tooker ME, O'Connell JR, Cole JB, Bickhart DM (2017) Selecting sequence variants to improve genomic predictions for dairy cattle. Gen Sel Evol 49:32. doi:10.1186/s12711-017-0307-4
59. Aguilar I, Legarra A, Tsuruta S, Misztal I (2013) Genetic evaluation using unsymmetric single step genomic methodology with large number of genotypes Interbull Bull 47:222-225
60. Aguilar I, Misztal I, Legarra A, Tsuruta S (2011) Efficient computation of the genomic relationship matrix and other matrices used in single-step evaluation. J Anim Breed Genet 128:422-428
61. Wiggans GR (2013) Current status of genomic evaluation for U.S. dairy cattle. In: China Emerging Markets Program Seminar.
62. Misztal I, Legarra A, Aguilar I (2014) Using recursion to compute the inverse of the genomic relationship matrix. J Dairy Sci 97:3943-3952
63. Misztal I (2016) Inexpensive computation of the inverse of the genomic relationship matrix in populations with small effective population size. Genetics 202:401-409. doi:10.1534/genetics.115.182089
64. Fragomeni BO, Lourenco DAL, Tsuruta S, Masuda Y, Aguilar I, Legarra A, Lawlor TJ, Misztal I (2015) Use of genomic single-step genomic BLUP with a large number of genotypes. J Dairy Sci 98:4090-4094
65. Masuda Y, Misztal I, Tsuruta S, Legarra A, Aguilar I, Lourenco DAL, Lawlor TJ (2016) Implementation of genomic recursions in single-step genomic best linear unbiased predictor for US Holsteins with a large number of genotyped animals. J Dairy Sci 99:1968-1974. doi:https://doi.org/10.3168/jds.2015-10540
66. Ostersen T, Christensen OF, Madsen P, Henryon M (2016) Sparse single-step method for genomic evaluation in pigs. Genet Sel Evol 48:48

67. Pocrnic I, Lourenco DAL, Chen CY, Herring WO, Misztal I (2019) Crossbred evaluations using single-step genomic BLUP and algorithm for proven and young with different sources of data. J Anim sci Preprint
68. Pocrnic I, Lourenco DAL, Masuda Y, Legarra A, Misztal I (2016) The dimensionality of genomic information and its effect on genomic prediction. Genetics 203:573-581. doi:10.1534/genetics.116.187013
69. Pocrnic I, Lourenco DAL, Masuda Y, Misztal I (2016) Dimensionality of genomic information and performance of the Algorithm for Proven and Young for different livestock species. Genet Sel Evol 48:82
70. Stranden I, Mantysaari EA (2014) Comparison of some equivalent equations to solve single-step GBLUP. Paper presented at the 10th World Congress on Gen. Appl. to Livest. Prod., Vancouver, Canada, 17-22 Aug 2014
71. Masuda Y, Misztal I, Legarra A, Tsuruta S, Lourenco DAL, Fragomeni BO, Aguilar I (2017) Technical note: Avoiding the direct inversion of the numerator relationship matrix for genotyped animals in single-step genomic BLUP solved with preconditioned conjugate gradient. J Anim Sci 95:49-52
72. Masuda Y, VanRaden PM, Misztal I, Lawlor TJ (2018) Differing genetic trend estimates from traditional and genomic evaluations of genotyped animals as evidence of preselection bias in US Holsteins. J Dairy Sci 101:5194-5206
73. Quaas RL, Pollak EJ (1981) Modified equations for sire models with groups. J Dairy Sci 64:1868–1872
74. Westell RA, Quaas RL, Vleck LDV (1988) Genetic Groups in an Animal Model. J Dairy Sci 71:1310-1318
75. Tsuruta S, Misztal I, Lourenco DAL, Lawlor TJ (2014) Assigning unknown parent groups to reduce bias in genomic evaluations of final score in US Holsteins. J Dairy Sci 97:5814–5821
76. Misztal I, Vitezica ZG, Legarra A, Aguilar I, Swan AA (2013) Unknown-parent groups in single-step genomic evaluation. J Anim Breed Genet 130:252-258
77. Matilainen K, Koivula M, Strandeen I, Aamand GP, Mantysaari EA (2016) Managing genetic groups in single-step genomic evaluations applied on female fertility traits in Nordic Red dairy cattle. Interbull Bull 50:71–75
78. Masuda Y, Misztal I, VanRaden PM, Lawlor TJ Genomic predictability of single-step GBLUP for production traits in US Holstein. In: ADSA Annual Meeting, Knoxville, TN, 2018. J. Dairy Sci. ,
79. Kennedy BW (1991) CR Henderson: The unfinished legacy. J Dairy Sci 74:4067-4081
80. VanRaden PM (1992) Accounting for inbreeding and crossbreeding in genetic evaluation of large populations. J Dairy Sci 75:3136-3144
81. Legarra A, Christensen OF, Vitezica ZG, Aguilar I, Misztal I (2015) Ancestral relationships using metafounders: Finite ancestral populations and across population relationships. Genetics 200:455-468
82. Christensen OF (2012) Compatibility of pedigree-based and marker-based relationship matrices for single-step genetic evaluation. Genet Sel Evol 44:37
83. Garcia-Baccino, C. A. AL, Christensen OF, Misztal I, Pocrnic I, Vitezica ZG, Cantet. RJC (2017) Metafounders are related to Fst fixation indices and reduce bias in single-step genomic evaluations. Genet Sel Evol 49:34
84. Meyer K, Tier B, Swan A (2018) Estimates of genetic trend for single-step genomic evaluations. Genet Sel Evol 50:39