genes

MDPI

Review

Single-Step Genomic Evaluations from Theory to Practice: Using SNP Chips and Sequence Data in BLUPF90

Daniela Lourenco [1,*], Andres Legarra [2], Shogo Tsuruta [1], Yutaka Masuda [1], Ignacio Aguilar [3] and Ignacy Misztal [1]

1 Department of Animal and Dairy Science, University of Georgia, Athens, GA 30602, USA; shogo@uga.edu (S.T.); yutaka@uga.edu (Y.M.); ignacy@uga.edu (I.M.)
2 Institut National de la Recherche Agronomique, UMR1388 GenPhySE, 31326 Castanet Tolosan, France; andres.legarra@inra.fr
3 Instituto Nacional de Investigación Agropecuaria (INIA), 11500 Montevideo, Uruguay; iaguilar@inia.org.uy
* Correspondence: danilino@uga.edu

Received: 19 June 2020; Accepted: 6 July 2020; Published: 14 July 2020

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 best linear unbiased prediction (ssGBLUP) and uses single nucleotide polymorphism (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. In this review, we focus solely on ssGBLUP. 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, number of phenotypes, and number of genotyped animals. Single-step GBLUP from the BLUPF90 software suite has been used for genomic evaluations worldwide. In this review, we will show theoretical developments and numerical examples of ssGBLUP using SNP data from regular chips to sequence data.

Keywords: genomic selection; genomic prediction; genome-wide association; single-step genomic BLUP

1. Introduction

In the early 1980s, Soller et al. [1] hypothesized that DNA markers like RFLPs (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 (SNPs). 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 three billion nucleotides in the bovine genome, and there are over 30 million SNPs or one every 100 nucleotides

is a SNP. Another reason is that SNP genotyping became automatized, relatively cheap, efficient (most loci are read) and highly reproducible (e.g., across laboratories), contrary to microsatellites.

In 2001, Meuwissen et al. [5] envisioned that genomic information could help animal breeders to generate more accurate breeding values, if a dense assay that covers the entire genome become available. Extending the idea of incorporating marker information into best linear unbiased prediction (BLUP), introduced by Fernando et al. [6] and extended to the whole genome by Lande et al. [7] and Haley et al. [8], 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 (DGVs) based on joint analyses of genotypes and phenotypes, method that was easily modified to use pseudo-phenotypes (i.e., estimated breeding values or EBVs adjusted for parent average and accuracy; or progeny deviations) only for genotyped animals as bulls. Following the same line, VanRaden [9] proposed an equivalent method called genomic BLUP (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 cost is greatly reduced (only selection candidates and highly represented animals such as bulls are genotyped), the traditional BLUP evaluation is kept unchanged and genomic selection can be carried out by using additional analyses. However, the multi-step method has some disadvantages: (a) DGVs 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 are cumbersome to obtain and may rely on accuracy obtained via approximated algorithms [10].

Although multistep methods were largely implemented for genomic evaluations worldwide, starting from 2009, this class of methods was not going 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 [11,12], and it was acknowledged that multi-step methods would eventually lead to bias predictions because BLUP predictions would ignore the effects of genomic selection [13]. Intending to solve these problems and to reduce the burden in obtaining genomic predictions, Misztal et al. [14] 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 is referred to as the **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. [15] 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. Aguilar et al. [16] and Christensen et al. [17] finally showed that although **H** is quite complex, its inverse is rather simple. This development was the landmark for the implementation of ssGBLUP in livestock populations. After 10 years, ssGBLUP has become the preferred tool for genomic evaluation and selection in many livestock species, namely beef cattle [18], pigs [19,20], broilers [21,22], layers [23], dairy sheep and goat [24], meat sheep [25], and fish [26]. Although ssGBLUP adds simplicity to the genomic evaluation system, its implementation involves several details and requires knowledge about peculiarities of the method. In this review, 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. Although the focus is on BLUPF90, we recognize there are other packages available for computing BLUP-based predictions with and without genomic information. Examples are ASREML [27], Wombat [28], Mix99 [29], DMU [30], MTG2 [31], GCTA [32], among others.

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 1997 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 supplied by the user [33].

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`), large-scale genetic evaluations using linear models (`blup90iod`) and linear-threshold models (`cblup90iod`), and accuracy approximation (`accf90`). For information on how to download and use the programs, check Appendix A.

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 Appendix B).

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, 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 Appendix C). All the programs, except 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 genotyped animals; however, this threshold is frequently raised. Additionally, all software in the BLUPF90 family is under constant development, where the main objective is to improve methods and computing performance. New updates on BLUPF90 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 SNPs 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 that the shared alleles come from the same ancestor in a base population.

Now we detail the ideas from VanRaden [9]. Assuming a matrix of SNPs inherited by each animal (M), with dimension $n \times m$ where n is the number of animals and m the number of SNPs. Several

parametrizations exist, but if AA = 0, AB = 1, and BB = 2, M has to be centered by allele frequency. Assuming a vector p with elements equal to p_i, the frequency of allele B at locus *i*:

$$\mathbf{Z} = \mathbf{M} - 2\mathbf{p}' \tag{1}$$

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 (non-centered) 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 $\mathrm{Var}(u) = \mathrm{E}\left(u^2\right) - \mathrm{E}(u)^2$ [34]. 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 *ma* deviated from the population mean [34], we obtain Equations (2) and (3):

$$u_\mathrm{i} = m_\mathrm{i}a - 2p'a = (m_\mathrm{i} - 2p')a = z_\mathrm{i}a \tag{2}$$

$$u_\mathrm{j} = m_\mathrm{j}a - 2p'a = (m_\mathrm{j} - 2p')a = z_\mathrm{j}a \tag{3}$$

According to Legarra et al. [34], if $\mathrm{Var}(\mathbf{a}) = \mathbf{I}\sigma_a^2$, or marker variance, and the genetic variance in Hardy–Weinberg equilibrium is $2\sum p_iq_i\sigma_a^2$, the rules of variances and covariances can be applied:

$$\mathrm{Cov}\left(u_\mathrm{i}, u_\mathrm{j}\right) = (z_\mathrm{i} - 2p)a(z_\mathrm{j} - 2p)a = (z_\mathrm{i} - 2p)(z_\mathrm{j} - 2p)\sigma_a^2 \tag{4}$$

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

$$\mathrm{Cov}\left(u_\mathrm{i}, u_\mathrm{j}\right) = z_\mathrm{i}z_\mathrm{j}\sigma_a^2 \tag{5}$$

Dividing the covariance by the genetic variance $2\sum p_iq_i\sigma_a^2$, 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 **u** is:

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

$$\mathrm{Var}(\mathbf{u}) = \mathbf{Z}\,\mathrm{Var}(\mathbf{a})\,\mathbf{Z}' \tag{7}$$

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

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:

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

$$\mathrm{Var}(\mathbf{u}) = \frac{\mathbf{ZZ}'}{2\sum_{i=1}^{SNP} p_i(1-p_i)}\sigma_u^2 \tag{10}$$

Therefore, and according to VanRaden [9], the genomic relationship (**G**) is given by:

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

then,

$$\mathrm{Var}(\mathbf{u}) = \mathbf{G}\sigma_u^2 \tag{12}$$

Therefore, genomic relationships are standardized covariances. When $\mathbf{ZZ}'$ is divided by $2\sum p_i(1-p_i)$, **G** becomes analogous to the pedigree 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. [35] and Amin et al. [36].

If **G** is centered using observed allele frequencies, the average over all elements is zero and average diagonal is 1 when there is no inbreeding. However, it is only when base allele frequencies are used that elements of **G** can be interpreted as elements of **A** (this will be more detailed later). 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 SNPs, or if there are similar individuals (e.g., clones), **G** becomes singular; therefore, cannot be inverted. To overcome this problem, usually, **G** is "blended" with a small percentage of an identity matrix or the pedigree relationship matrix among genotyped animals ($\mathbf{A}_{22}$):

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

where the blending parameter α is usually 95% but can vary from 99 to 80% [37], or even to 50% [38]. The blending parameter $(1-\alpha)$ can be understood as the fraction of genetic variance not explained by markers and computed by maximum likelihood methods (see below).

2.3. From GBLUP to ssGBLUP

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

The GBLUP is equivalent to SNP-BLUP, but in GBLUP genomic breeding values ($\mathbf{u} = \mathbf{Za}$) are estimated, instead of SNP effects (**a**) in SNP-BLUP. It also assumes that SNPs have a priori a normal distribution; the majority of SNPs have a small effect, and very few have moderate to large effect. Using a simple animal model as shown in (14) and (15):

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Wu} + \boldsymbol{e} \tag{14}$$

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

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 $\mathbf{u} \sim N\left(0, \mathbf{G}\sigma_u^2\right)$.

Therefore, GBLUP is a BLUP where **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:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Wu} + \mathbf{Wg} + \mathbf{e} \tag{16}$$

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 ($\mathbf{u}_g$) becomes

$$\mathbf{u}_g = \mathbf{u} + \mathbf{g} \tag{17}$$

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

Therefore,

$$\mathbf{G} = \alpha\mathbf{G} + (1-\alpha)\mathbf{A}_{22} \tag{19}$$

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 [17]. When $(1-\alpha)$ is used strictly to make **G** (semi-) positive definite, it is called a 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 (DGVs); (5) blending DGVs with average of parent's EBV, which is known as parent average (PA), with published EBV, or with PTA. The main issue on having an evaluation with several steps is that some errors and biases can be introduced during those steps [10], and that BLUP will not be robust to genomic selection decisions [13].

The idea for ssGBLUP came from the fact that usually 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. [15] 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 [15,17]:

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

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

If we consider that

$$\text{Var}(\mathbf{u}) = \mathbf{A}\sigma_u^2 \tag{22}$$

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

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

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

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 [39]. 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:

$$\begin{aligned} \text{Var}(\mathbf{u}_1) &= \text{Var}\left(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2 + \boldsymbol{\varepsilon}\right) \\ &= \text{Var}(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2) + \text{Var}(\boldsymbol{\varepsilon}) \\ &= \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} + \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} \end{aligned} \tag{25}$$

Rearranging:

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

Therefore,

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

$$\mathrm{Var}(\mathbf{u}_2) = \mathrm{Var}(\mathbf{Z}\mathbf{a}) = \mathbf{G} \quad (27)$$

$$\mathrm{Cov}(\mathbf{u}_1, \mathbf{u}_2) = \mathrm{Cov}\left(\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{u}_2, \mathbf{u}_2\right) = \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathrm{Var}(\mathbf{u}_2) \quad (28)$$

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

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

$$\mathbf{H} = \begin{pmatrix} \mathrm{Var}(\mathbf{u}_1) & \mathrm{Cov}(\mathbf{u}_1, \mathbf{u}_2) \\ \mathrm{Cov}(\mathbf{u}_2, \mathbf{u}_1) & \mathrm{Var}(\mathbf{u}_2) \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{G} - \mathbf{A}_{22})\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{G} \\ \mathbf{G}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{G} \end{pmatrix} \quad (30)$$

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

which can be simplified to:

$$\mathbf{H} = \mathbf{A} + \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix}[\mathbf{G} - \mathbf{A}_{22}]\begin{bmatrix} \mathbf{I} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (32)$$

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 [16,17]:

$$\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}_{22}^{-1} \end{bmatrix} \quad (33)$$

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:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{W}\mathbf{u} + \mathbf{e} \quad (34)$$

The MME for ssGBLUP becomes:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{W} \\ \mathbf{W}'\mathbf{X} & \mathbf{W}'\mathbf{W} + \mathbf{H}^{-1}\frac{\sigma_e^2}{\sigma_u^2} \end{bmatrix}\begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathrm{y} \\ \mathbf{W}'\mathrm{y} \end{bmatrix} \quad (35)$$

The distribution of **u** becomes:

$$\mathbf{u} \sim N\left(\mathbf{0}, \mathbf{H}\sigma_u^2\right) \quad (36)$$

Therefore, the only difference between BLUP and ssGBLUP is that A^{-1} is replaced by $\mathbf{H}^{-1}$. Subsequently, all tools based on BLUP mixed model equations, as the restricted maximum likelihood

(REML [40]), can be easily 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 [10], relatively higher accuracy than multistep methods [41–45], potential to account for pre-selection bias as all pedigree, phenotypic, and genomic information can be included in the model [12,13], and 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 [46]. 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 three columns: animal, sire, and dam. The phenotype file is named phenotypes.txt and contains animal, sex, phenotype, true breeding value, and 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 and SNP_genotype. The last file (gen_map.txt) contains the map for SNPs: SNP identification, chromosome number, position (in base pairs).

After running `renumf90` to renumber the data (see Appendix B), 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 Appendix B for a description of keywords and values). The following command line can be used to save the screen output to a file:

`blupf90` renf90.par | `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 map_file gen_map.txt
```

Preconditioner conjugate gradient [47] 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 output file provided by `blupf90` with solutions for all effects is the "solutions" file, 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 Appendix B).

Box 2. First five lines of the `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	−0.05317279

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 in R.

```
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)
```

The `blupf90` software outputs a large amount of information on the screen, including quality control checks, statistics for $\mathbf{G}$ 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 Appendix C.

2.5. Compatibility between Pedigree and Genomic Relationships

Based on how $\mathbf{H}$ is constructed, the central element is $\mathbf{G} - \mathbf{A}_{22}$ (see Equation (31)), which implies both matrices should be compatible [10,48]. Compatibility can be understood as both matrices referring to the same genetic base and to the same genetic variance. However, genomic relationships can be biased if $\mathbf{G}$ is constructed based on allele frequencies other than the ones from the base population [9]. However, 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. [39], these estimates are not very accurate because the base population is several generations away from the genotyped individuals. Additionally, in certain

contexts such as missing pedigrees there is not a uniquely defined base population. 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 [9]. However, if the population is under selection, mean breeding values should change from the base population to the genotyped individuals (i.e., they should 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. [48] 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 under selection. It can be calculated as:

$$\rho = \frac{1}{n^2}\left(\sum_i \sum_j \mathbf{A}_{22\ i,j} - \sum_i \sum_j \mathbf{G}_{i,j}\right) \tag{37}$$

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

$$\mathbf{G}* = (1 - \rho/2)\,\mathbf{G} + 11'\,\rho \tag{38}$$

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 by implicitly fitting a constant μ, unlike in Hsu et al. [49] where the constant is fit explicitly.

When ssGBLUP was first implemented [16] in the BLUPF90 family of programs, $\mathbf{A}^{-1}$ was constructed based on Henderson [50] and Quaas [51] assuming no inbreeding, a frequently-used approximation [52,53], $\mathbf{G}^{-1}$ was constructed based on VanRaden [9], and $\mathbf{A}_{22}^{-1}$ was based on Colleau [54] and fully considered inbreeding. As the algorithms to construct $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$ implicitly consider inbreeding, but not the algorithm for A^{-1}, $\mathbf{H}^{-1}$ 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 [16,55]:

$$\mathbf{H}^{-1} = \mathbf{A}^{-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tau\mathbf{G}^{-1} - \omega\mathbf{A}_{22}^{-1} \end{bmatrix} \tag{39}$$

Primarily, ω controls inflation due to incompleteness of pedigree and τ controls additive genetic variance [56]. 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 observed that when inbreeding was considered in $\mathbf{A}^{-1}$ by adding an extra option to the `renumf90` parameter file (see below), the need for ω lower than 1 was reduced. It was rather surprising that ignoring inbreeding in the set-up of $\mathbf{A}^{-1}$, which is harmless in BLUP applications, had such a great impact in ssGBLUP. In fact, when genotyped animals have complete pedigree, τ and ω are likely to be 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 while keeping GEBV with an acceptable level of inflation/deflation. To ensure consideration of inbreeding in the set-up of $\mathbf{A}^{-1}$ the lines

```
INBREEDING
pedigree
```

need to be included in the parameter file for `renumf90`, and then the genetic effect in the parameter file for `blupf90` needs to be

```
RANDOM_TYPE
add_an_upginb
```

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 $1-\alpha$ (or β) equal 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. [57]: $\mathbf{G}^{*} = \varphi\mathbf{G} + \delta$, where $\varphi = \left[\frac{\overline{diagA_{22}} - \overline{offdiagA_{22}}}{\overline{diagG} - \overline{offdiagG}}\right]$ and $\delta = \overline{diagA_{22}} - \overline{diagG} * \varphi$. To change the adjustment of **G** to the one proposed by Vitezica et al. [48] and demonstrated in Equation (38), the following option can be added to the `blupf90` parameter file (e.g., renf90.par):

```
OPTION tunedG 4
```

A total of four different adjustments are implemented in the BLUPF90 family of programs; however, types 2 [57] and 4 [48] 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 $\omega = \tau = 1$. 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 [9] as $\mathbf{u} = \mathbf{Za}$ and $\mathrm{Var}(\mathbf{u}) = \mathrm{Var}(\mathbf{Za})$. Using this idea, the selection index equation for GBLUP can be represented by:

$$\hat{\mathbf{u}} = \mathbf{G}\left[\mathbf{G} + \mathbf{R}\left(\frac{\sigma_a^2}{\sigma_e^2}\right)\right]^{-1}\left(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\right) \tag{40}$$

where **R** is a diagonal matrix with (heterogeneous if needed) residual variance. If $\hat{\mathbf{u}}\middle|\hat{\mathbf{a}} = \mathbf{Z}\hat{\mathbf{a}}$, replacing the first **G** by **Z**′, weighted by the ratio of SNP to additive direct variances (i.e., $k = \sigma_a^2/\sigma_u^2$), would allow the calculation of SNP effects (a) [9]:

$$\hat{\mathbf{a}} = \mathbf{Z}'k\left[\mathbf{G} + \mathbf{R}\left(\frac{\sigma_a^2}{\sigma_e^2}\right)\right]^{-1}\left(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\right) \tag{41}$$

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:

$$\mathbf{w} = \left[\mathbf{G} + \mathbf{R}\left(\frac{\sigma_u^2}{\sigma_e^2}\right)\right]^{-1}\left(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\right) \tag{42}$$

then,

$$\hat{\mathbf{a}} = k\mathbf{Z}'\mathbf{w} \tag{43}$$

therefore,

$$\hat{\mathbf{u}} = \mathbf{G}\mathbf{w} \tag{44}$$

In this way,

$$\mathbf{w} = \mathbf{G}^{-1}\hat{\mathbf{u}} \tag{45}$$

Finally, the SNP effects can be calculated as in (46):

$$\hat{\mathbf{a}} = k\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}} \tag{46}$$

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

$$\hat{\mathbf{a}}\middle|\hat{\mathbf{u}} = k\mathbf{D}\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}} \tag{47}$$

Thus, given GEBV from ssGBLUP are available, SNP effects are calculated as [58]:

$$\hat{\mathbf{a}} = k\mathbf{D}\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}} \tag{48}$$

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 [18]. Except for the small portion of remaining pedigree variation, they are identical to GEBV [18]. 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 that **A** is poorly constructed for them and results in incompatibilities with **G** (which is correct) [56]. Additionally, if lots of animals are genotyped, say weekly, but they do not contribute to the evaluation (as in young animals that do not have phenotype or progeny yet), having IP for them reduces computing costs.

Another feature of having SNP effects is the ability to account for the fact that SNPs explain different proportion of genetic variance on the trait, and this leads to iterative methods similar to Bayesian regressions (i.e., BayesA, BayesB). An iterative method was proposed to add different weights for SNPs under ssGBLUP, which is called weighted ssGBLUP [58]. In this method, seven steps are needed:

1. Set the diagonal matrix of SNP variance or weight as an identity, **D = I**
2. Compute the genomic relationships: $\mathbf{G} = \mathbf{Z}\mathbf{D}\mathbf{Z}'/k$, 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: $\hat{\mathbf{a}} = k\mathbf{D}\mathbf{Z}'\mathbf{G}^{-1}\hat{\mathbf{u}}$
5. Estimate SNP variance for SNP i e.g., as $d_i = a_i^2$ (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 one to two rounds. Different formulas can be used to calculate SNP variance, but all of them are approximations. Several authors have reported decrease in GEBV accuracy and increase in bias over iterations [59,60] 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 [9] proposed and successfully applied in US dairy cattle a formula to calculate SNP variance that limits the change over iterations, avoiding extreme values. This method is called non-linearA:

$$d_i = \text{CT}^{\frac{|\hat{a}_i|}{\sigma(\hat{\mathbf{a}})}-2} \tag{49}$$

where CT is a constant that determines the departure from normality; $|\hat{a}_i|$ is the absolute estimated SNP effect for marker i, and $\sigma\left(\hat{\mathbf{a}}\right)$ is the standard deviation of the vector of estimated SNP effects.

Garcia et al. [26] and Fragomeni et al. [61] 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 [9], meaning the distribution for SNP effects approaches a t distribution with large degrees of freedom, e.g., approaching a normal distribution.

Considering SNP variances 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., >10 k genotyped animals) [60], even for less polygenic traits. If the data allows to accurately estimate SNP effects, there is no advantage in selecting SNPs 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 SNPs per trait are easy to implement for single- but not multiple-trait models [62].

Once the variance for each SNP is calculated, the proportion of additive genetic variance can be plotted for all SNPs 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).

More formally, and according to the common use in ambitious GWAS studies, p-values for SNPs can be calculated as [63–65]:

$$pval_i = 2\left(1 - \Phi\left(\left|\frac{\hat{a}_i}{sd(\hat{a}_i)}\right|\right)\right) \quad (50)$$

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 [65]:

$$var(\hat{a}_i) = k\ \alpha\ b\ \mathbf{z}_i'\mathbf{G}^{-1}(\mathbf{G}\sigma_u^2 - \mathbf{C}^{u_2u_2})\mathbf{G}^{-1}\mathbf{z}_i\ b\ \alpha\ k \quad (51)$$

where $\mathbf{C}^{u_2u_2}$ is the portion of the inverse of the LHS of MME for ssGBLUP (34) referent to genotyped animals; b is (1- ρ/2), which is a function of the tuning parameter in (38); k and α were defined previously.

2.8. Using postGSf90 to Compute SNP Effects, SNP Variances, and p-Values

If the objective is to backsolve GEBV to SNP effect and then calculate variance explained by SNPs, `postGSf90` can be used. This software was primarily developed to serve this purpose, but recently was modified to also compute p-values for SNPs [63]. As this software relies on GEBV to calculate SNP effect and variance, `blupf90` needs to be run first with three additional options in renf90.par:

```
OPTION saveGInverse
OPTION saveA22Inverse
OPTION snp_p_value
```

The first and second options save $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$, respectively, and the third option saves $\mathbf{C}^{u_2u_2}$, all 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 saveA22Inverse
OPTION snp_p_value
```

The first two options are to read $\mathbf{G}^{-1}$ and $\mathbf{A}_{22}^{-1}$, respectively; the third option is used now to calculate p-values. A fourth option (i.e., `OPTION windows_variance` x) can be used if variance for SNP is to be calculated based on windows of x SNPs instead of individual SNP [58]. Based on Equation (48), `postGSf90` needs GEBV, SNP content, and 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 file 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, and chrsnp_pval, which are used to generate Manhattan plots for SNP effect, proportion of variance explained by *n* adjacent SNPs, 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 $d_i = 2p_i\ (1-p_i)\ a_i^2$. However, four 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, and computing *p*-value is in Box 6.

Box 6. Parameter file for running `postGSf90`using non-linearA variance 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 map_file gen_map.txt
OPTION readGInverse
OPTION readA22Inverse
OPTION snp_p_value
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 [63]. To check how to have weighted ssGBLUP where SNP effect, SNP variance, and GEBV are updated in an iterative way, see Appendix D.

2.9. Accounting for Sequence Variants in ssGBLUP

Genomic selection relies on linkage disequilibrium (LD) between SNPs and quantitative trait nucleotides (QTNs). By having dense SNP panels (i.e., >50,000 SNP), it is more likely that a 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 SNPs the accuracy of genomic selection will increase. VanRaden et al. [66] showed an average increase of 1.6% in reliability of GEBV for a simulated trait when using 500,000 instead of 50,000 SNPs. According to Meuwissen et al. [67], the ideal SNP density is given by whole-genome sequence data. As millions of SNPs are screened, the causative variants are expected to be among them. However, the use of high-density SNP chips did not increase accuracy from medium-size chips, and use of sequence data is showing only marginally higher accuracies.

Using simulated data, Fragomeni et al. [68] 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. [61] tested the performance of ssGBLUP when using nearly 54,000 SNPs and when adding 17,000 significant variants discovered in a GWAS (pre-selected sequence SNPs) that involved 33 traits [69]. Although VanRaden et al. [69] 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. [61] 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 SNPs. Although causative variants can be included in ssGBLUP assuming different weights for SNPs, maximizing the accuracy of GEBV would require the true identification of all causative variants, their substitution effect, their position, and the proportion of additive genetic variance they explain. To identify some of the causative variants, a large number of sequenced animals with phenotypes is needed. When a large amount of information is available, the accuracy may be high enough; therefore, improvements from the incorporation of causative variants are likely to be small for large data sets. When the number of genotyped animals is larger than the number of independent chromosome segments, the accuracy is maximized without SNP weighting/selection [70–72].

An optimal algorithm for finding causative SNPs would be to assign a large variance to those causative SNPs while reducing the variance of nearby SNPs or setting it to 0. This would resemble methods that perform a sequential estimation of SNP effects/variances. In such methods, SNPs close to causative variants are automatically disregarded. Although there are three different weighting methods implemented in BLUPF90 programs, any external weight can be considered, including the ones from Bayesian regressions. The only requirement is that those external weights have to be rescaled to sum to the number of SNPs used in the model. However, Gualdron-Duarte et al. [73] found that improvements in predictivity from unweighted GBLUP to BayesR and other methods were similar to improvements from unweighted ssGBLUP to weighted ssGBLUP with external weights. To check how to consider different weights or variance for causative variants in ssGBLUP, see Appendix D.

The default configuration of `preGSf90`, `blupf90`, and `postGSf90` assume a maximum number of SNPs equal to 400,000. In the presence of sequence data (or over 400,000 SNPs), an extra option is required:

```
OPTION maxsnp x
```

where x is the new maximum number of SNPs.

Although the most common way to include pre-selected sequence SNPs is to add them to the current SNP panel, it is possible to have an analysis where both are considered as separate components. In such a case, there is a need to fit two animal effects into the model—one for the current SNP panel and one for the pre-selected sequence SNPs [74]. However, the gains in accuracy using GBLUP with two animal effects was limited [74,75]. There are no reports on the literature about the use of SNPs from a panel and pre-selected from sequence fitting two **H** in ssGBLUP. Accommodating two **G** (GBLUP) or two **H** (ssGBLUP) is possible using BLUPF90 software, although this may have the same impact on accuracy as using different weights for pre-selected sequence SNPs. Additionally, a strong assumption of no correlation between the two random animal effects has to be assumed.

To accommodate two genomic matrices in blupf90, the inverse of those two matrices should be constructed separately using `preGSf90` and saved to a file. The `preGSf90` has an option to save $\mathbf{H}^{-1}$ but not **H**, because the latter is never constructed:

```
OPTION saveHinv
```

This option saves the diagonals and upper diagonals of $\mathbf{H}^{-1}$ as a plain text file (Hinv.txt) in the format of row, column, and value, where row and column are based on renumbered IDs. When using `blupf90`, the random type should be set as `user_file` (see `RANDOM_TYPE` in Appendix B) and the file name has to be provided (e.g., Hinv_chip.txt, Hinv_seq.txt, ...). The random type `user_file` should be used as many times as the number of different SNP sets were used to compute $\mathbf{H}^{-1}$(Box 7).

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.

Box 7. Parameter file for running `blupf90` using two $\mathbf{H}^{-1}$.

```
DATAFILE
renf90.dat
NUMBER_OF_TRAITS
     1
NUMBER_OF_EFFECTS
     3
OBSERVATION(S)
  1
WEIGHT(S)

EFFECTS:                    POSITIONS_IN_DATAFILE                 NUMBER_OF_LEVELS
TYPE_OF_EFFECT[EFFECT NESTED]
2      2 cross
3    12010 cross
3    12010 cross
RANDOM_RESIDUAL VALUES
0.60000
RANDOM_GROUP
   2
RANDOM_TYPE
user_file
FILE
Hinv_chip.txt
(CO)VARIANCES
0.30000
RANDOM_GROUP
  3
RANDOM_TYPE
user_file
FILE
Hinv_seq.txt
(CO)VARIANCES
0.10000
```

2.10. Large-Scale Genomic Evaluations with ssGBLUP

The most expensive operation in ssGBLUP, as implemented in Aguilar et al. [16] and Christensen et al. [17], 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 [76,77]. 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 780,000 (Steve Miller, 2020; personal communication) and the US dairy industry has already collected over 3.4 M 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 [78].

To overcome the limitation set by the number of genotyped animals in ssGBLUP, Misztal et al. [79] 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 [50] and Quaas [51] to recursively construct the inverse of $\mathbf{A}$. The logic behind the construction of $\mathbf{G}^{-1}_{\text{APY}}$ 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 ($\mathbf{u}_c$):

$$\mathbf{u}_n = \mathbf{P}_{nc}\mathbf{u}_c + \boldsymbol{\Psi}_n \tag{52}$$

where $\mathbf{P}_{nc}$ is a matrix that relates breeding values for noncore to core animals, and $\mathbf{\Psi}_n$ is a diagonal matrix with estimation errors. Following further developments [80], $\mathbf{G}^{-1}_{\text{APY}}$ can be constructed as:

$$\mathbf{G}^{-1}_{\text{APY}} = \begin{bmatrix} \mathbf{G}^{-1}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{G}^{-1}_{cc}\mathbf{G}_{cn} \\ \mathbf{I} \end{bmatrix} \mathbf{M}^{-1}_{nn} \begin{bmatrix} -\mathbf{G}_{nc}\mathbf{G}^{-1}_{cc} & \mathbf{I} \end{bmatrix} \tag{53}$$

with $m_{nnn_{ii}} = g_{ii} - \mathrm{g}_{ic}\mathbf{G}^{-1}_{cc}\mathrm{g}_{ci}$. The APY algorithm creates a generalized sparse inverse of $\mathbf{G}$ at approximately a linear cost in computing and storage [79,80] and has been extensively tested for beef cattle [18], dairy cattle [81,82], and pigs [83,84]. This algorithm enables ssGBLUP evaluations with millions of genotyped animals, as the only inverse needed is for the core animals. Pocrnic et al. [85] and Pocrnic et al. [86] 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 of 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. [86] found that the minimum number of core animals was 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 et al. [87] and Masuda et al. [88] proposed to compute an efficient inverse indirectly as a product of sparse matrices:

$$\mathbf{A}^{-1}_{22} = \mathbf{A}^{22} - \mathbf{A}^{21}\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12} \tag{54}$$

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 $\mathbf{A}^{-1}_{22}$ for 570,000 genotyped animals extracted from a population of 10 M animals was around 11 min [88]. Single-step GBLUP with $\mathbf{G}^{-1}_{\text{APY}}$ and efficient $\mathbf{A}^{-1}_{22}$ was successfully applied to over 10.9 M cows with milking records, 13.5M animals in the pedigree, and about 2.3 M genotyped Holsteins [89]; using 15,000 core animals, the complete evaluation for a model with 18 type traits took four and a half days to converge. Within this time stamp, the construction of $\mathbf{G}^{-1}_{\text{APY}}$ and $\mathbf{A}^{-1}_{22}$ took one day. In fact, this time depends on the total number of genotyped animals and the core group size.

Unfortunately, the subroutines to create $\mathbf{G}^{-1}_{\text{APY}}$ and efficient $\mathbf{A}^{-1}_{22}$ are not implemented in the free distribution of BLUPF90 family of programs.

2.11. Unknown Parent Groups (UPG) and Metafounders in ssGBLUP

Commercial populations, especially sheep, beef and dairy cattle, often have incomplete pedigrees. In BLUP, missing parents are modeled by UPG [51,90,91]. 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 be estimated accurately and to avoid confounding with other effects in the model [51]. In ssGBLUP, when UPG are applied only to pedigree relationships, the convergence rate can be slow [92]. Misztal et al. [93] 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 [90]:

$$\mathbf{H}^{-1*} = \mathbf{A}^{-1*} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}^{-1}_{22} & -\left(\mathbf{G}^{-1} - \mathbf{A}^{-1}_{22}\right)\mathbf{Q}_2 \\ \mathbf{0} & -\mathbf{Q}'_2\left(\mathbf{G}^{-1} - \mathbf{A}^{-1}_{22}\right) & \mathbf{Q}'_2\left(\mathbf{G}^{-1} - \mathbf{A}^{-1}_{22}\right)\mathbf{Q}_2 \end{bmatrix} \tag{55}$$

where $\mathbf{Q}_2$ is a matrix that relates genotyped animals to groups; $\mathbf{G}^{-1}$ can be replaced by $\mathbf{G}^{-1}_{APY}$ 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 [94]. Revised UPGs also worked well for the US Holstein data up to 2014 [56]. However, using data updated to 2015, Masuda et al. [95] 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 $\mathbf{G}^{-1}$, as genomic relationships are not affected by missing pedigree, implying UPG are automatically accounted for. Tsuruta et al. [96] showed that UPG for $\mathbf{G}^{-1}$ were not estimable for young genotyped animals in an 18-type trait genomic evaluation.

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 [97,98]. Legarra et al. [99] 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. [48]. Based on the metafounders theory, $\mathbf{G}$ would be derived using 0.5 allele frequencies as an "absolute reference" [100], and $\mathbf{A}$ would be scaled for compatibility with $\mathbf{G}$ using covariances among and within metafounders. According to Legarra et al. [99] 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 [101]. In simulations and real data, the concept of metafounders delivered the least biased predictions [38,101]. In ssGBLUP, $\mathbf{H}^{-1}$ with metafounders ($\mathbf{H}^{\Gamma-1}$) can be represented by:

$$\mathbf{H}^{\Gamma-1} = \mathbf{A}^{\Gamma-1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} - \mathbf{A}^{\Gamma-1}_{22} \end{bmatrix} \quad (56)$$

where $\mathbf{\Gamma}$ is the relationship matrix among metafounders, which can be estimated using generalized least squares [101]. Once $\mathbf{\Gamma}$ is inverted, Henderson [50] rules can be used to construct the inverse of the pedigree relationship matrix. As metafounders are based on UPG, estimating $\mathbf{\Gamma}$ strongly depends on how the groups are assigned. Estimating $\mathbf{\Gamma}$ is an active area of research as many UPGs are weakly related to genotyped individuals.

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 Appendix B). 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
```

To set UPG only for $\mathbf{A}^{-1}$ and $\mathbf{A}^{-1}_{22}$, a second option is needed to remove UPG for $\mathbf{G}^{-1}$:

```
OPTION TauOmegaQ2 0.0 1.0
```

As the metafounders concept is still recent, the BLUPF90 developers are currently working on a standalone software to estimate $\mathbf{\Gamma}$, which is called `gammaf90`. 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. [101] to estimate $\mathbf{\Gamma}$ 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 Appendix B).

3. Conclusions

The BLUPF90 is a complete software suite for the most common computations needed in animal breeding and genetics. The programs are highly optimized and have been under constant development for over 20 years. Single-step GBLUP, which is one of the main tools for genomic analysis, was first implemented in 2009. Since then, several changes were made to make ssGBLUP flexible to any model, number of traits, number of phenotypes, number of genotyped animals, and sequence data. Single-step GBLUP is fully supported in the BLUPF90 software suite and has been used for genomic evaluations worldwide.

Author Contributions: Conceptualization, D.L.; software development, A.L., S.T., Y.M., I.A. and I.M.; writing—original draft preparation, D.L.; writing—review and editing, A.L., S.T., Y.M., I.A. and I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This project was primarily supported by grants from American Angus Association, Holstein Association USA (Brattleboro, VT), Zoetis, Cobb-Vantress, Pig Improvement Company, Smithfield Premium Genetics, the US Department of Agriculture's National Institute of Food and Agriculture (Agriculture Computations in genomic selection and Food Research Initiative competitive grant 2015-67015-22936), and the European Unions' Horizon 2020 Research & Innovation programme under grant agreement N 772787 -SMARTER.

Acknowledgments: This paper is based on numerous studies done by several people from our animal breeding and genetics group at the University of Georgia, including former and current postdocs and graduate students. Two anonymous reviewers are thanked for helpful suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A.

Appendix A.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 (Linux, OSX Mac, or Windows);
(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 the 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:

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

The above command works for Linux and Mac (in Windows if modern terminals like MobaXterm are used) and will provide the parameter file when `blupf90` asks for it and will save the screen output to a file named out.log

Appendix A.2. Downloading a Toy Dataset

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 using QMSim [46]. Files with pedigree, phenotypes, and genotypes are available at https://github.com/danielall/Data_ssGBLUP. The pedigree file is named pedigree.txt and contains three columns: animal, sire, and dam. The phenotype file is named phenotypes.txt and contains animal, sex, phenotype, true breeding value, and 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 and SNP_genotype. The last file (gen_map.txt) contains the map for SNPs: SNP identification, chromosome number, position (in base pairs).

Appendix B.

Renumbering the Data with renumf90:

The BLUPF90 software suite only works with numeric entries (i.e., integer or real) and levels of all effects need to be consecutive starting from 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, removing (pruning) uninformative animals.

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 (Table A1). If there is no need to use FIELDS_PASSED TO OUTPUT and WEIGHT(S), simply put an empty line as a value.

Table A1. Possible values and descriptions for the keywords used in `renumf90`.

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 in matrix form
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 in Tables A2 and A3:

Table A2. Declaration of fixed effects in `renumf90`.

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

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 number 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. Table A3 has 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):

Table A3. Declaration of random effects in `renumf90` and all associated keywords.

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 in the pedigree file are passed to the new pedigree file. If no input, the default value is 3
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 $\mathbf{A}^{-1}$ 'pedigree' = calculated from pedigree 'file_with_inb.txt' to provide a file with two columns: animal ID and inbreeding coefficient
(CO)VARIANCES	(Co)variance components for general random effects in matrix form
(CO)VARIANCES_PE	(Co)variance components for permanent environmental effect in matrix form
(CO)VARIANCES_MPE	(Co)variance components for maternal permanent environmental effect in matrix form
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 in `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 five 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 map_file 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:

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

Hint 4: `OPTION map_file` 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 map file should contain a header indicating the columns for the SNP identification (SNP_ID), chromosome (CHR), and the position of the SNP in base pairs (POS).

The renumbered output files and contents are:

(1) renf90.dat—is the renumbered phenotype file and contains three columns: phenotype, renumbered sex code, and 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) Three 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 into three columns: code, number of observations, and 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 map_file gen_map.txt
```

The parameter file generated by `renumf90` is also based on keywords and values that are described in Table A4:

Table A4. Keywords in values in the new parameter file created by `renumf90`.

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, and 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 two 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 twos 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 map_file	Need to be followed by the name of the SNP map file when available
OPTION	Any extra option that the BLUPF90 family of programs can take. To see other options, check the online manual

Appendix C.

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 SNPs with low minor allele frequency (MAF) and monomorphic,

SNPs departing from the Hardy–Weinberg Equilibrium, and SNPs 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 SNPs 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 the inverses. The options are `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 map_file gen_map.txt
OPTION saveCleanSNPs
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:

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

If a SNP map file is provided, three new files are generated by `preGSf90`: genotypes.txt_clean, geotypes.txt_XrefID_clean, and gen_map.txt_clean, which contain the information after removing SNPs 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 map_file 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 $\mathbf{H}$ is not needed in ssGBLUP, this matrix cannot be created using `preGSf90`.

Appendix D.

Iterative Weighted ssGBLUP with `blupf90` *and* `postGSf90`*:*

Variance or weights for SNPs can be used to construct the genomic relationship matrix when a diagonal matrix of weights is included in the equation to create $\mathbf{G}$ [9], as $\mathbf{G} = \frac{\mathbf{ZDZ}'}{2\sum p_i(1-p_i)}$. If $\mathbf{G}$ can be updated with weights, the fact SNPs 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 [60], whereas small genomic data may produce extreme SNP variances.

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 [26,61]. Assuming the data is renumbered (see Appendix B) 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 [61]. The parameter file for `blupf90` should include extra options to save $\mathbf{G}^{-1}$ (`OPTION saveGInverse`) and $\mathbf{A}_{22}^{-1}$ (`OPTION saveA22Inverse`), to use weights for SNPs (`OPTION weightedG` weights.txt), and to avoid quality control (`OPTION no_quality_control`). Avoiding quality control in the iterative run reduces computing time and keeps the number of SNPs constant in consecutive iterations. The file weights.txt contains a column with weights for SNPs, where the number of lines is the number of SNPs. In the first iteration, this file contains a column of ones. 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 map_file gen_map.txt
OPTION saveGInverse
OPTION saveA22Inverse
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:

```
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 map_file gen_map.txt
OPTION readGInverse
OPTION readA22Inverse
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 seven 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. If there is a divergence of variances, the first iteration should be used (e.g., no weights). If there is a desire to use external weights, they should be used instead of the column seven of snp_sol. The external weights have to be rescaled to sum to the number of SNPs used in the model.

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 of weighted ssGBLUP may be a sign of misidentified or low-quality genomic samples.

References

1. Soller, M.; Beckmann, J.S. Genetic polymorphism in varietal identification and genetic improvement. *Theor. Appl. Genet.* **1983**, *67*, 25–33. [CrossRef] [PubMed]
2. The International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature* **2001**, *409*, 928–933. [CrossRef] [PubMed]
3. Stonecking, M. From the evolutionary past. *Nature* **2001**, *409*, 821–822. [CrossRef] [PubMed]
4. Schork, N.J.; Fallin, D.; Lanchbury, S. Single nucleotide polymorphisms and the future of genetic epidemiology. *Clin. Genet.* **2000**, *58*, 250–264. [CrossRef]
5. Meuwissen, T.H.E.; Hayes, B.J.; Goddard, M.E. Prediction of total genetic value using genome-wide dense marker maps. *Genetics* **2001**, *157*, 1819–1829.
6. Fernando, R.L.; Grossman, M. Marker-assisted selection using best linear unbiased prediction. *Genet. Sel. Evol.* **1989**, *21*, 467–477. [CrossRef]
7. Lande, R.; Thompson, R. Efficiency of marker-assisted selection in the improvement of quantitative traits. *Genetics* **1990**, *124*, 743–756.
8. Haley, C.S.; Vischer, P.M. Strategies to utilize marker-quantitative trait loci associations. *J. Dairy Sci.* **1998**, *81*, 85–97. [CrossRef]
9. VanRaden, P.M. Efficient methods to compute genomic predictions. *J. Dairy Sci.* **2008**, *91*, 4414–4423. [CrossRef]
10. Legarra, A.; Chistensen, O.F.; Aguilar, I.; Misztal, I. Single step, a general approach for genomic selection. *Livest. Prod. Sci.* **2014**, *166*, 54–65. [CrossRef]
11. Wiggans, G.R.; Cooper, T.A.; VanRaden, P.M.; Cole, J.B. Technical note: Adjustment of traditional cow evaluations to improve accuracy of genomic predictions. *J. Dairy Sci.* **2011**, *94*, 6188–6193. [CrossRef] [PubMed]
12. Wiggans, G.R.; VanRaden, P.M.; Cooper, T.A. Technical note: Adjustment of all cow evaluations for yield traits to be comparable with bull evaluations. *J. Dairy Sci.* **2012**, *95*, 3444–3447. [CrossRef]
13. Patry, C.; Ducrocq, V. Evidence of biases in genetic evaluations due to genomic preselection in dairy cattle. *J. Dairy Sci.* **2011**, *94*, 1011–1020. [CrossRef]
14. Misztal, I.; Legarra, A.; Aguilar, I. Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information. *J. Dairy Sci.* **2009**, *92*, 4648–4655. [CrossRef] [PubMed]
15. Legarra, A.; Aguilar, I.; Misztal, I. A relationship matrix including full pedigree and genomic information. *J. Dairy Sci.* **2009**, *92*, 4656–4663. [CrossRef] [PubMed]
16. Aguilar, I.; Misztal, I.; Johnson, D.L.; Legarra, A.; Tsuruta, S.; Lawlor, T.J. Hot topic: A unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. *J. Dairy Sci.* **2010**, *93*, 743–752. [CrossRef] [PubMed]
17. Christensen, O.F.; Lund, M.S. Genomic prediction when some animals are not genotyped. *Genet. Sel. Evol.* **2010**, *42*, 2. [CrossRef]
18. Lourenco, D.A.L.; Tsuruta, S.; Fragomeni, B.O.; Masuda, Y.; Aguilar, I.; Legarra, A. Genetic evaluation using single-step genomic best linear unbiased predictor in American Angus. *J. Anim. Sci.* **2015**, *93*, 2653–2662. [CrossRef]
19. Forni, S.; Aguilar, I.; Misztal, I. Different genomic relationship matrices for single-step analysis using phenotypic, pedigree, and genomic information. *Genet. Sel. Evol.* **2011**, *43*, 1. [CrossRef]
20. Lourenco, D.A.L.; Tsuruta, S.; Fragomeni, B.O.; Chen, C.Y.; Herring, W.O.; Misztal, I. Crossbreed evaluations in single-step genomic best linear unbiased predictor using adjusted realized relationship matrices. *J. Anim. Sci.* **2016**, *94*, 909–919. [CrossRef]
21. Chen, C.Y.; Misztal, I.; Aguilar, I.; Tsuruta, S.; Meuwissen, T.H.E.; Aggrey, S.E. Genome-wide marker-assisted selection combining all pedigree phenotypic information with genotypic data in one step: An example using broiler chickens. *J. Anim. Sci.* **2011**, *89*, 23–28. [CrossRef] [PubMed]

22. Lourenco, D.A.L.; Fragomeni, B.O.; Tsuruta, S.; Aguilar, I.; Zumbach, B.; Hawken, R.J. Accuracy of estimated breeding values with genomic information on males, females, or both: An example in broiler chicken. *Genet. Sel. Evol.* **2015**, *47*, 56. [CrossRef]
23. Yan, Y.; Wu, G.; Liu, A.; Sun, C.; Han, W.; Li, G. Genomic prediction in a nuclear population of layers using single-step models. *Poult. Sci.* **2018**, *97*, 397–402. [CrossRef] [PubMed]
24. Rupp, R.; Mucha, S.; Larroque, H.; McEwan, J.; Conington, J. Genomic application in sheep and goat breeding. *Anim. Front.* **2016**, *6*, 39–44. [CrossRef]
25. Brown, D.J.; Swan, A.A.; Boerner, V.; Li, L.; Gurman, P.M.; McMillan, A.J. Single-Step Genetic Evaluations in the Australian Sheep Industry. In Proceedings of the 11th World Congress on Genetics Applied to Livestock Production, Auckland, New Zealand, 11–16 February 2018.
26. Garcia, A.L.S.; Bosworth, B.; Waldbieser, G.; Misztal, I.; Tsuruta, S.; Lourenco, D.A.L. Development of genomic predictions for harvest weight and carcass weight in channel catfish. *Genet. Sel. Evol.* **2018**, *50*, 66. [CrossRef]
27. Gilmour, A.R.; Gorgel, B.J.; Cullis, B.R.; Thompson, R. *ASReml User Guide Release 2.0*; VSN International: Hemel Hempstead, UK, 2006.
28. Meyer, K. WOMBAT—A tool for mixed model analyses in quantitative genetics by restricted maximum likelihood (REML). *J. Zhejiang Univ. Sci. B* **2007**, *8*, 815–821. [CrossRef] [PubMed]
29. Lidauer, M.; Matilainen, K.; Mantysaari, E.; Pitkanen, T.; Taskinen, M.; Stranden, I. *Technical Reference Guide for MiX99 Solver*; Natural Resources Institute Finland: Jokioinen, Finland, 2015.
30. Madsen, P.; Jensen, J.; Labouriau, R.; Christensen, O.F.; Sahana, G. DMU—A Package for Analyzing Multivariate Mixed Models in quantitative Genetics and Genomics. In Proceedings of the 10th World Congress of Genetics Applied to Livestock Production, Vancouver, BC, Canada, 17–22 August 2014.
31. Lee, S.H.; Van der Werf, J.H. MTG2: An efficient algorithm for multivariate linear mixed model analysis based on genomic information. *Bioinformatics* **2016**, *32*, 1420–1422. [CrossRef]
32. Yang, J.; Lee, S.H.; Goddard, M.E.; Visscher, P.M. GCTA: A tool for genome-wide complex trait analysis. *Am. J. Hum. Genet.* **2011**, *88*, 76–82. [CrossRef]
33. Aguilar, I.; Tsuruta, S.; Masuda, Y.; Lourenco, D.A.L.; Legarra, A.; Misztal, I. BLUPF90 suite of programs for animal breeding with focus on genomics. In Proceedings of the 11th World Congress on Genetics Applied to Livestock Production, Auckland, New Zealand, 11–16 February 2018.
34. Legarra, A.; Lourenco, D.A.L.; Vitezica, Z. Bases for Genomic Predictions. 2018. Available online: http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=gsip.pdf (accessed on 30 March 2020).
35. Leutenegger, A.-L.; Prum, B.; Génin, E.; Verny, C.; Lemainque, A.; Clerget-Darpoux, F. Estimation of the inbreeding coefficient through use of genomic data. *Am. J. Hum. Genet.* **2003**, *73*, 516–523. [CrossRef] [PubMed]
36. Amin, N.; van Duijn, C.M.; Aulchenko, Y.S. A genomic background based method for association analysis in related individuals. *PLoS ONE* **2007**, *2*, e1274. [CrossRef]
37. Guarini, A.R.; Lourenco, D.A.L.; Brito, L.F.; Sargolzaei, M.; Baes, C.F.; Miglior, F. Comparison of genomic predictions for lowly heritable traits using multi-step and single-step genomic best linear unbiased predictor in Holstein cattle. *J. Dairy Sci.* **2019**, *101*, 8076–8086. [CrossRef]
38. Meyer, K.; Tier, B.; Swan, A. Estimates of genetic trend for single-step genomic evaluations. *Genet. Sel. Evol.* **2018**, *50*, 39. [CrossRef]
39. Gengler, N.; Mayeres, P.; Szydlowski, M. A simple method to approximate gene content in large pedigree populations: Application to the myostatin gene in dual-purpose Belgian Blue cattle. *Animal* **2007**, *1*, 21–28. [CrossRef] [PubMed]
40. Patterson, H.D.; Thompson, R. Recovery of inter-block information when block sizes are unequal. *Biometrika* **1971**, *58*, 545–554. [CrossRef]
41. Baloche, G.; Legarra, A.; Salle, G.; Larroque, H.; Astruc, J.M.; Robert-Granie, C. Assessment of accuracy of genomic prediction for French Lacaune dairy sheep. *J. Dairy Sci.* **2014**, *97*, 1107–1116. [CrossRef] [PubMed]
42. Christensen, O.F.; Madsen, P.; Nielsen, B.; Ostersen, T.; Su, G. Single-step methods for genomic evaluation in pigs. *Animal* **2012**, *6*, 1565–1571. [CrossRef] [PubMed]
43. Gray, K.A.; Cassady, J.P.; Huang, Y.; Maltecca, C. Effectiveness of genomic prediction on milk flow traits in dairy cattle. *Genet. Sel. Evol.* **2012**, *44*, 24. [CrossRef]

44. Lourenco, D.A.L.; Misztal, I.; Tsuruta, S.; Aguilar, I.; Ezra, E.; Ron, M. Methods for genomic evaluation of a relatively small genotyped dairy population and effect of genotyped cow information in multiparity analyses. *J. Dairy Sci.* **2014**, *97*, 1742–1752. [CrossRef]
45. Tsuruta, S.; Misztal, I.; Lawlor, T.J. Short communication: Genomic evaluations of final score for US Holsteins benefit from the inclusion of genotypes on cows. *J. Dairy Sci.* **2013**, *96*, 3332–3335. [CrossRef]
46. Sargolzaei, M.; Schenkel, F.S. QMSim: A large-scale genome simulator for livestock. *Bioinformatics* **2009**, *25*, 680–681. [CrossRef]
47. Tsuruta, S.; Misztal, I.; Stranden, I. Use of the preconditioned conjugate gradient algorithm as a generic solver for mixed-model equations in animal breeding applications. *J. Anim. Sci.* **2001**, *79*, 1166–1172. [CrossRef] [PubMed]
48. Vitezica, Z.G.; Aguilar, I.; Misztal, I.; Legarra, A. Bias in genomic predictions for populations under selection. *Genet. Res. (Camb.)* **2011**, *93*, 357–366. [CrossRef] [PubMed]
49. Hsu, W.L.; Garrick, D.J.; Fernando, R.L. The Accuracy and Bias of Single-Step Genomic Prediction for Populations Under Selection. *G3* **2017**, *7*, 2685–2694. [CrossRef] [PubMed]
50. Henderson, C.R. A simple method for computing the inverse of a relationship matrix used in prediction of breeding values. *Biometrics* **1976**, *32*, 69–83. [CrossRef]
51. Quaas, R.L. Additive genetic model with groups and relationships. *J. Dairy Sci.* **1988**, *71*, 1338–1345. [CrossRef]
52. Golden, B.L.; Brinks, J.S.; Bourdon, R.M. A performance programmed method for computing inbreeding coefficients from large data sets for use in mixed-model analyses. *J. Anim. Sci.* **1991**, *69*, 3564–3573. [CrossRef] [PubMed]
53. Mehrabani-Yeganeh, H.; Gibson, J.P.; Schaeffer, L.R. Including coefficients of inbreeding in BLUP evaluation and its effect on response to selection. *J. Anim. Breed. Genet.* **2000**, *117*, 145–151. [CrossRef]
54. Colleau, J.J. An indirect approach to the extensive calculation of relationship coefcients. *Genet. Sel. Evol.* **2002**, *34*, 409–421. [CrossRef]
55. Tsuruta, S.; Misztal, I.; Aguilar, I.; Lawlor, T.J. Multiple-trait genomic evaluation of linear type traits using genomic and phenotypic data in US Holsteins. *J. Dairy Sci.* **2011**, *94*, 4198–4204. [CrossRef]
56. Misztal, I.; Bradford, H.L.; Lourenco, D.A.L.; Tsuruta, S.; Masuda, Y.; Legarra, A. Studies on Inflation of GEBV in Single-Step GBLUP for Type. *Interbull Bull.* **2017**, *51*, 38–42.
57. Chen, C.Y.; Misztal, I.; Aguilar, I.; Legarra, A.; Muir, W.M. Effect of different genomic relationship matrices on accuracy and scale. *J. Anim. Sci.* **2011**, *89*, 2673–2679. [CrossRef] [PubMed]
58. Wang, H.; Misztal, I.; Aguilar, I.; Legarra, A.; Muir, W.M. Genome-wide association mapping including phenotypes from relatives without genotypes. *Genet. Res.* **2012**, *94*, 73–83. [CrossRef] [PubMed]
59. Lee, J.; Cheng, H.; Garrick, D.; Golden, B.; Dekkers, J.; Park, K. Comparison of alternative approaches to single-trait genomic prediction using genotyped and non-genotyped Hanwoo beef cattle. *Genet. Sel. Evol.* **2017**, *49*, 2. [CrossRef] [PubMed]
60. Lourenco, D.A.L.; Fragomeni, B.O.; Bradford, H.L.; Menezes, I.R.; Ferraz, J.B.S.; Tsuruta, S. Implications of SNP weighting on single-step genomic predictions for different reference population sizes. *J. Anim. Breed. Genet.* **2017**, *134*, 463–471. [CrossRef]
61. Fragomeni, B.O.; Lourenco, D.A.L.; Legarra, A.; VanRaden, P.M.; Misztal, I. Alternative SNP weighting for single-step genomic best linear unbiased predictor evaluation of stature in US Holsteins in the presence of selected sequence variants. *J. Dairy Sci.* **2019**, *102*, 10012–10019. [CrossRef]
62. Tiezzi, F.; Maltecca, C. Accounting for trait architecture in genomic predictions of US Holstein cattle using a weighted realized relationship matrix. *Genet. Sel. Evol.* **2015**, *47*, 24. [CrossRef]
63. Aguilar, I.; Legarra, A.; Cardoso, F.; Masuda, Y.; Lourenco, D.; Misztal, I. Frequentist p-values for large-scale single step genome-wide association, with an application to birth weight in American Angus. *Genet. Sel. Evol.* **2019**, *51*, 28. [CrossRef]
64. Bernal-Rubio, Y.L.; Gualdron-Duarte, J.L.; Bates, R.O.; Ernst, C.W.; Nonneman, D.; Rohrer, G.A. Meta-analysis of genome-wide association from genomic prediction models. *Anim. Genet.* **2015**, *47*, 36–48. [CrossRef]
65. Gualdron-Duarte, J.L.; Cantet, R.J.C.; Bates, R.O.; Ernest, C.W.; Raney, N.E.; Steibel, J.P. Rapid screening for phenotype-genotype associations by linear transformations of genomic evaluations. *BMC Bioinform.* **2014**, *15*, 246. [CrossRef]

66. VanRaden, P.M.; O'Connell, J.R.; Wiggans, G.R.; Weigel, K.A. Genomic evaluations with many more genotypes. *Genet. Sel. Evol.* **2011**, *43*, 10. [CrossRef]
67. Meuwissen, T.H.E.; Hayes, B.; Goddard, M. Genomic selection: A paradigm shift in animal breeding. *Anim. Front.* **2016**, *6*, 6–14. [CrossRef]
68. Fragomeni, B.O.; Lourenco, D.A.L.; Masuda, Y.; Misztal, I. Incorporation of Causative Quantitative Trait Nucleotides in Single-step GBLUP. *Genet. Sel. Evol.* **2017**, *49*, 59. [CrossRef] [PubMed]
69. VanRaden, P.M.; Tooker, M.E.; O'Connell, J.R.; Cole, J.B.; Bickhart, D.M. Selecting sequence variants to improve genomic predictions for dairy cattle. *Genet. Sel. Evol.* **2017**, *49*, 32. [CrossRef]
70. Daetwyler, H.D.; Pong-wong, R.; Villanueva, B.; Woolliams, J.A. The impact of genetic architecture on genome-wide evaluation methods. *Genetics* **2010**, *185*, 1021–1031. [CrossRef] [PubMed]
71. Karaman, E.; Cheng, H.; Firat, M.Z.; Garrick, D.J.; Fernando, R.L. Un upper bound for accuracy of prediction using GBLUP. *PLoS ONE* **2016**, *11*, e0161054. [CrossRef]
72. Pocrnic, I.; Lourenco, D.; Masuda, Y.; Misztal, I. Accuracy of genomic BLUP when considering a genomic relationship matrix based on number of largest eigenvalues—A simulation study. *Genet. Sel. Evol.* **2019**, *51*, 75. [CrossRef] [PubMed]
73. Gualdron-Duarte, J.L.; Gori, A.S.; Hubin, X.; Lourenco, D.; Charlier, C.; Misztal, I. Application of the Adaptive MultiBLUP strategy for genomic predictions in Belgian Blue Beef cattle. *BMC Genom.* **2020**. under review.
74. Brondum, R.F.; Su, G.; Janss, L.; Sahana, G.; Guldbrandtsen, B.; Boichard, D. Quantitative trait loci markers derived from whole genome sequence data increases the reliability of genomic prediction. *J. Dairy Sci.* **2015**, *98*, 4107–4116. [CrossRef]
75. Liu, A.; Lund, M.S.; Boichard, D.; Karaman, E.; Fritz, S.; Aamand, G.P. Improvement of genomic prediction by integrating additional single nucleotide polymorphisms selected from imputed whole genome sequencing data. *Heredity* **2020**, *124*, 37–49. [CrossRef]
76. Aguilar, I.; Legarra, A.; Tsuruta, S.; Misztal, I. Genetic evaluation using unsymmetric single step genomic methodology with large number of genotypes. *Interbull Bull.* **2013**, *47*, 222–225.
77. Aguilar, I.; Misztal, I.; Legarra, A.; Tsuruta, S. Efficient computation of the genomic relationship matrix and other matrices used in single-step evaluation. *J. Anim. Breed. Genet.* **2011**, *128*, 422–428. [CrossRef] [PubMed]
78. Wiggans, G.R. Current status of genomic evaluation for U.S. dairy cattle. In Proceedings of the China Emerging Markets Program Seminar, Holstein, Austrilia, 11–12 March 2013.
79. Misztal, I.; Legarra, A.; Aguilar, I. Using recursion to compute the inverse of the genomic relationship matrix. *J. Dairy Sci.* **2014**, *97*, 3943–3952. [CrossRef]
80. Misztal, I. Inexpensive computation of the inverse of the genomic relationship matrix in populations with small effective population size. *Genetics* **2016**, *202*, 401–409. [CrossRef]
81. Fragomeni, B.O.; Lourenco, D.A.L.; Tsuruta, S.; Masuda, Y.; Aguilar, I.; Legarra, A. Hot topic: Use of genomic recursions in single-step genomic best linear unbiased predictor (BLUP) with a large number of genotypes. *J. Dairy Sci.* **2015**, *98*, 4090–4094. [CrossRef] [PubMed]
82. Masuda, Y.; Misztal, I.; Tsuruta, S.; Legarra, A.; Aguilar, I.; Lourenco, D.A.L. 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.* **2016**, *99*, 1968–1974. [CrossRef] [PubMed]
83. Ostersen, T.; Christensen, O.F.; Madsen, P.; Henryon, M. Sparse single-step method for genomic evaluation in pigs. *Genet. Sel. Evol.* **2016**, *48*, 48. [CrossRef]
84. Pocrnic, I.; Lourenco, D.A.L.; Chen, C.Y.; Herring, W.O.; Misztal, I. Crossbred evaluations using single-step genomic BLUP and algorithm for proven and young with different sources of data. *J. Anim. sci.* **2019**, *97*, 1513–1522. [CrossRef] [PubMed]
85. Pocrnic, I.; Lourenco, D.A.L.; Masuda, Y.; Legarra, A.; Misztal, I. The dimensionality of genomic information and its effect on genomic prediction. *Genetics* **2016**, *203*, 573–581. [CrossRef]
86. Pocrnic, I.; Lourenco, D.A.L.; Masuda, Y.; Misztal, I. Dimensionality of genomic information and performance of the Algorithm for Proven and Young for different livestock species. *Genet. Sel. Evol.* **2016**, *48*, 82. [CrossRef]
87. Stranden, I.; Mantysaari, E.A. Comparison of some equivalent equations to solve single-step GBLUP. In Proceedings of the 10th World Congress of Genetics Applied to Livestock Production, Vancouver, BC, Canada, 17–22 August 2014.

88. Masuda, Y.; Misztal, I.; Legarra, A.; Tsuruta, S.; Lourenco, D.A.L.; Fragomeni, B.O. 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.* **2017**, *95*, 49–52.
89. Tsuruta, S.; Lawlor, T.J.; Lourenco, D.A.L.; Misztal, I. Bias in genomic predictions by mating practices for linear type traits in a large-scale genomic evaluation. *J. Dairy Sci* **2020**. under review.
90. Quaas, R.L.; Pollak, E.J. Modified equations for sire models with groups. *J. Dairy Sci.* **1981**, *64*, 1868–1872. [CrossRef]
91. Westell, R.A.; Quaas, R.L.; Vleck, L.D.V. Genetic Groups in an Animal Model. *J. Dairy Sci.* **1988**, *71*, 1310–1318. [CrossRef]
92. Tsuruta, S.; Misztal, I.; Lourenco, D.A.L.; Lawlor, T.J. Assigning unknown parent groups to reduce bias in genomic evaluations of final score in US Holsteins. *J. Dairy Sci.* **2014**, *97*, 5814–5821. [CrossRef] [PubMed]
93. Misztal, I.; Vitezica, Z.G.; Legarra, A.; Aguilar, I.; Swan, A.A. Unknown-parent groups in single-step genomic evaluation. *J. Anim. Breed. Genet.* **2013**, *130*, 252–258. [CrossRef] [PubMed]
94. Matilainen, K.; Koivula, M.; Strandeen, I.; Aamand, G.P.; Mantysaari, E.A. Managing genetic groups in single-step genomic evaluations applied on female fertility traits in Nordic Red dairy cattle. *Interbull Bull.* **2016**, *50*, 71–75.
95. Masuda, Y.; Misztal, I.; VanRaden, P.M.; Lawlor, T.J. Differing genetic trend estimates from traditional and genomic evaluations of genotyped animals as evidence of preselection bias in US Holsteins. *J. Dairy Sci.* **2018**, *101*, 5194–5206. [CrossRef]
96. Tsuruta, S.; Lourenco, D.A.L.; Masuda, Y.; Misztal, I.; Lawlor, T.J. Controlling bias in genomic breeding values for young genotyped bulls. *J. Dairy Sci.* **2019**, *102*, 9956–9970. [CrossRef]
97. Kennedy, B.W. CR Henderson: The unfinished legacy. *J. Dairy Sci.* **1991**, *74*, 4067–4081. [CrossRef]
98. VanRaden, P.M. Accounting for inbreeding and crossbreeding in genetic evaluation of large populations. *J. Dairy Sci.* **1992**, *75*, 3136–3144. [CrossRef]
99. Legarra, A.; Christensen, O.F.; Vitezica, Z.G.; Aguilar, I.; Misztal, I. Ancestral relationships using metafounders: Finite ancestral populations and across population relationships. *Genetics* **2015**, *200*, 455–468. [CrossRef] [PubMed]
100. Christensen, O.F. Compatibility of pedigree-based and marker-based relationship matrices for single-step genetic evaluation. *Genet. Sel. Evol.* **2012**, *44*, 37. [CrossRef] [PubMed]
101. Garcia-Baccino, C.A.A.L.; Christensen, O.F.; Misztal, I.; Pocrnic, I.; Vitezica, Z.G. Metafounders are related to Fst fixation indices and reduce bias in single-step genomic evaluations. *Genet. Sel. Evol.* **2017**, *49*, 34. [CrossRef] [PubMed]