MASTER INTERUNIVERSITARIO EN MEJORA GENÉTICA ANIMAL Y BIOTECNOLOGÍA DE LA REPRODUCCIÓN

EVALUATION OF THE POTENTIAL USE OF MULTIPLE POPULATIONS IN GENOMIC EVALUATION

Tesis de Master
Universidad Politecnica de Valencia, Julio 2013

Elena Flavia Mouresan

Director:
Luis Varona

INDEX

ABSTRACT

Marker-assisted selection has been introduced into the breeding programs of livestock populations during the last few years. Particularly, the cattle industry was the first to incorporate the Genomic Selection (GS) into its breeding programs. GS has been proved to be a very useful tool for increasing the efficiency of the breeding programs of dairy cattle but until now it has not given the expected results in beef cattle. The structure of the beef cattle industry consists of many small populations. The absence of a large reference population and the low predictive ability of one population over another prevent the obtainment of accurate predictions. Combining information of animals from different populations in order to construct a large, reference population has been proposed in order to increase the accuracy of the predictions for all populations.

In this study, 4 purebred populations were simulated. Two of these populations diverged under selection, each for a different trait, and the other two under random mating. From these purebred populations 6 admixed populations were created combining them in pairs and 1 admixed population that included data from all purebreds. The predictive ability of all purebred and admixed populations was evaluated under two scenarios of marker density, two scenarios of number of genes that control the traits and 3 scenarios of number of generations of divergence.

The results of this study show that training in purebred populations gives good accuracies when validating in the same population, although slightly lower, and declining with the generations, when the population is under selection for the trait of interest. Training in admixed populations gives higher accuracies for the breeds that participate in the training set than for the ones that do not, but these accuracies are significantly lower than those obtained from training in purebred population. Finally, when the training set is made of all 4 breeds, the accuracies are slightly better for all breeds than those obtained with the 2-breed admixed population.

When the marker density is higher the accuracies are better for all purebred and admixed populations but the admixed populations are more favored than the purebreds. The number of genes that control the traits seems to affect the predictions of the populations under selection but not the others.

RESUMEN

En los últimos años, la selección asistida por marcadores se ha introducido en los programas de mejora genética animal. En particular, el sector de vacuno de leche ha sido el primero en introducir la selección genómica (GS) en sus esquemas de mejora. En ellos, se ha probado que la selección genómica es una herramienta de gran interés. Por el contrario, su potencial utilización en vacuno de carne todavía no es evidente, debido a la presencia de varias poblaciones, la ausencia de una población de referencia grande y la baja capacidad predictiva de una población en otra. Como alternativa, se ha propuesto la creación de una metapoblación a partir de la información disponible en varias poblaciones, que permita incrementar la precisión en todas ellas.

En este trabajo, se han simulado 4 poblaciones puras partiendo de un único tronco evolutivo. Dos de estas poblaciones se separaron mediante selección, cada una para un carácter diferente, y las otras dos exclusivamente mediante deriva genética. A partir de estas 4 poblaciones, se generaron 6 poblaciones compuestas o metapoblaciones de dos poblaciones combinándolas por parejas y una metapoblación compuesta por las 4 subpoblaciones. La capacidad predictiva de todas las poblaciones puras y compuestas se evaluó bajo dos escenarios de densidad de marcadores y de número de genes y 3 escenarios de número de generaciones de divergencia.

Los resultados del trabajo mostraron que cuando se utilizó la información de las poblaciones puras se obtuvieron buenos resultados en la propias poblaciones, aunque decrecieron lentamente a medida que se incrementaron las generaciones de selección sobre el carácter de interés. Las predicciones obtenidas en las metapoblaciones fueron más altas sobre las poblaciones que participan en su formación, y mucho menores para las demás poblaciones. Finalmente, las predicciones a partir de la metapoblación compuesta de las cuatro poblaciones fueron ligeramente mejores que a partir de las compuestas de dos poblaciones.

Cuando la densidad de marcadores se incrementó, los resultados fueron mejores tanto para las poblaciones puras como para las compuestas, pero el incremento fue superior para las metapoblaciones. Por otra parte, el número de genes implicados en las poblaciones afectó solamente los resultados de las poblaciones seleccionadas.

INTRODUCTION

Anessential task in the development of breeding programs in livestock populations is the definition of objectives and criteria of selection. The objectives of selection are composed by traits that define the general goal for the breeding program, with the aim to obtain better individuals for production. The criteria of selection are the measurable traits that permit to identifythe best individualsin order to be used as reproductive individuals to generate the following generation. The simplest criteria of selection are defined by the information provided by only one trait and the methods of selection differ one from the other according to the sources of information used.

Mass selection

The first and most simple method was individual or mass selection (Falconer and Mackay, 1996). This method uses as criterion of selection exclusively the phenotypic performance of the candidates to selection. Thus, the best individuals are chosen as reproductive individuals assuming that a better phenotype reveals a better breeding value. The efficiency of this method depends largely on the heritability of the trait considered. Heritability is defined as the ratio of the additive variance and the phenotypic variance of a trait. It explains the proportion of the total phenotypic variability caused by the additive action of genes. Heritability´s most important aspect comes from its predictive function, since it represents the degree of reliability of the phenotypic value as an indicator of the breeding value. Individual selection is quite efficient when the heritability is high but gives poor results when it is low.

To overcome the obstacle of the low accuracy of the predicted breeding values caused by the low heritability of some traits, the mean performance of repeated records of the candidates can be used, when available, as criterion of selection. The justification for the use of this method is the assumption that all the phenotypic records of an individual are controlled by the same genetic and permanent environmental effects and that the variation observed is only produced by temporary environmental effects or residuals. The last ones are averaged when the mean performance of various records is used. Thus, when the residual variation is large (or heritability is low) the increase of accuracy with the number of recordsis very important. However, when the residual variance is low (or heritability is large), it provides barely any additional information. The main

disadvantage of this method is that it extends the generation interval that results in the reduction of the annual genetic gain.

The annual genetic gain is a measurement of the expected progress of a breeding program (Falconer and Mackay, 1996) and it is defined as:

$$\Delta G = \frac{i\rho\sigma_a}{L}$$

where ΔG is the expected genetic gain, i is the intensity of selection, ρ is the accuracy of the prediction, σ_a is the additive standard deviation and L is the generation interval. Intensity of selection (i) is a standardized measure of the difference between the average performance of the selected individuals with respect to the whole population. The sense of the standardization is to have a dimensionless parameter that can be compared between breeding programs. Accuracy of prediction (ρ) is the correlation coefficient between the predicted breeding value and the true breeding value. The additive standard deviation (σ_a) is the square root of the additive variance of the trait considered. Generation interval (L) is defined as the average age of both parents when progeny are born.

Index of selection

Later on, new statistical developments allowed the use of other sources of phenotypic information. Data from the relatives of the candidates were incorporated to the breeding programs with the purpose of increasing the accuracy of the predicted breeding values. The increase in the accuracy is a result of the fact that related individuals share genes and in many cases also the same environment (ex. piglets from the same litter or piglets from different litters that share a common sire). This resemblance between the related individuals can contribute with additional information over the genetic merit of the candidates. The types of relatives more frequently used are full-sibs or half-sibs, parents and progeny because the closer the genetic relatedness between the individuals, more genes they share and more new information is contributed to the prediction of the breeding value. For selection purposes, an index is built using all phenotypic information available from the candidate and its relatives (Lynch and Walsh,1998), weighted according to their degree of genetic relatedness. When the true variance components are used, this index is the best lineal prediction of an individual breeding value and its properties include the maximization of the correlation between the true

breeding value and the index (Lynch and Walsh, 1998). An important limitation to the use of the selection index is the need for the data to be adjusted for systematic effects before its application. These systematic effects are generally unknown and their estimation can result to be a difficult task, especially when no prior data exist. Further, it requires the inverse of the covariance matrix of the phenotypic observations that may be impossible to calculate for large data sets.

Nonetheless, the availability of data of the relatives made it possible to apply methods of selection for traits that are not expressed by the candidates to selection. Some examples of these cases are the prediction of the breeding value of males for milk production, the evaluation of individuals for meat quality traits and the selection for longevity. In all cases, the weighted average performance of the relatives is the criterion of selection.

Multiple traits

In animal production, the economic gain may depend on various productive and reproductive traits. Thus, the objective and the criteria of selection can be composed by several traits and the total genetic merit of an individual is not only the additive value for one trait but the sum of the additive values of several traits adjusted by economic weights. The economic weight of a trait represents the increase in the economic gain due to the genetic improvement of a single unit of the trait when the mean of the rest of the traits does not change (Weller, 1994).

The selection index methodology allows the use of information of several traits to estimate the total genetic merit of the candidates, from its own information and from its relatives. In fact, the index is calculated as the sum of phenotypic records of various traits weighted by regression coefficients for each individual. The goal is to obtain those regression coefficients that allow the index to be the minimum quadratic estimator of the total genetic merit. The accuracy of the index is affected by the economic weights, the phenotypic and genetic parameters that are not known and have to be estimated previously.

Other alternatives for multiple trait selection are tandem selection (Falconer and MacKay, 1996), where one generation is devoted for selection for one trait and the following one for another trait, and selection by independent levels, where minimum limits of performance are established for every trait and the candidates have to overpass

them in order to be selected. The advantage of both methods is that they are easier to apply since not all animals' records are required. The efficiency of the individual index is always better because it uses all the information available, while the efficiency of the other two methods depends on the genetic characteristics of the traits under selection.

Best Linear Unbiased Predictor (BLUP)

As an extension of the selection index theory, Henderson (1949) developed the methodology named Best Lineal Unbiased Prediction (BLUP). This methodology permits the estimation of systematic effects and the prediction of breeding values simultaneously. With known variances components, the properties of BLUP include the maximization of the correlation between the true and predicted breeding value. Under BLUP, the predictors are lineal functions of observations and the estimations are unbiased ($E(a|\hat{a}) = \hat{a}$). Later on, the same author also presented the mixed-model equations that give the estimations of systematic effects and the predictions of the breeding values without the need to calculate the inverse of the observation matrix (Henderson, 1950). The mixed-model equations are:

$$\begin{bmatrix} \boldsymbol{X`R^{-1}X} & \boldsymbol{X`R^{-1}Z} \\ \boldsymbol{Z´R^{-1}X} & \boldsymbol{Z`R^{-1}Z + G^{-1}} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{b}} \\ \widehat{\boldsymbol{a}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X`R^{-1}y} \\ \boldsymbol{Z`R^{-1}y} \end{bmatrix}$$

Where

$\boldsymbol{y}$ is a vector of n observations,

$\widehat{\boldsymbol{b}}$ is a vector of estimates of p levels of systematic effects,

$\widehat{\boldsymbol{a}}$ is a vector of predictions of q breeding values,

$\boldsymbol{X}$ is a design matrix $n \times p$ which relates records to systematic effects, and $\boldsymbol{X}$` is it´s transposed matrix,

$\boldsymbol{Z}$ is a design matrix $n \times q$ which relates records to breeding values, and $\boldsymbol{Z}$` is it´s transposed matrix,

$\boldsymbol{R^{-1}}$ is the inverse of the residual variance-covariance matrix,

$\boldsymbol{G^{-1}}$ is the inverse of the additive genetic variance-covariance matrix.

Since $\boldsymbol{R^{-1}}$ is an identity matrix because residual effects are independent one from the other, it can be factorized from both sides of the equation to give the equivalent mixed-model equations:

$$\begin{bmatrix} \boldsymbol{X`X} & \boldsymbol{X`Z} \\ \boldsymbol{Z`X} & \boldsymbol{Z`Z + A^{-1}\alpha} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{b}} \\ \widehat{\boldsymbol{a}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X`y} \\ \boldsymbol{Z`y} \end{bmatrix}$$

With $\alpha = \sigma_e^2/\sigma_\alpha^2$ and $\boldsymbol{A^{-1}}$ being the inverse of numerator relationship matrix which indicates the additive genetic relationship among individuals.

Computing the numerator relationship matrix ($\boldsymbol{A}$) and inverting it can be very difficult for the large data sets that are normally used for the genetic evaluations. Fortunately, Henderson (1976) proposed a recursive method to calculate the $\boldsymbol{A}$ matrix and its inverse just from the pedigree, when inbreeding is ignored. To take account of the effect of inbreeding, Henderson's (1976) approach needs the diagonal elements of the relationship matrix to calculate the $\boldsymbol{A^{-1}}$ matrix. Later on, Quaas (1976) generalized the procedure and allowed the fast computation of inbreeding coefficients or the construction of the $\boldsymbol{A^{-1}}$ matrix, without setting up the relationship matrix first.

The use of the mixed-model equations, for the prediction of breeding values and the estimation of systematic effects, has been extended to several models, such as:

Sire Model

The sire model was applied in the dairy cattle industry for the evaluation of the sires for milk production. In the Sire model, only the genetic effects of the sires are included in the model and the progeny records are used for the evaluation. However, the genetic merit of the dams is not accounted for as it is assumed that all mates are of similar genetic merit. The advantage of this model is that the number of equations is reduced compared to the animal model.

Animal Model

This model includes as random effects the additive values of all individuals in the population, regardless the number of systematic effects that have to be considered, and the residuals. It is the model more widely applied for several traits and animal species. It assumes that the additive variance and the residual variance are independent.

Repeatability Model

This model is applied when multiple records from an individual are available. It includes as random effects the additive effects and the permanent effects. It assumes that the genetic correlation and the environmental correlation between the pairs of records for each individual equals to 1. It predicts not only breeding values but also permanent environmental effects.

Reduced animal model

The reduced animal model (Quaas and Pollak, 1980) was developed to reduce the computational burden of the full animal model. This procedure allows setting up equations only for the parents and the breeding values of the progeny are obtained by back-solving from the parental breeding values.

Animal model with groups

The animal model with groups of unknown parents (Westell and Van Vleck, 1988) avoids assuming the same average breeding value for all the founders. Thus, they must be grouped according to the year of birth and genetic origin, and a genetic mean of breeding values is estimated for each group.

Model with common environment effects

The resemblance between relatives is produced by sharing genes and also by sharing common environment. This model includes as random effects and predicts the common environmental effects apart from the breeding values. It is used mostly in species with large families such as pigs and chickens.

Multivariate Animal Model

The multivariate Animal Model (Henderson and Quass, 1976) expands the mixed model methodology to several traits. As a consequence, the additive and residual (co-) variance matrices between traits are involved in the mixed model equations.

Animal Model with Maternal Effects

Some traits are affected by the individual´s genes but also by environmental effects that are controlled by the genes of the mother. In the animal model with maternal effects (Quaas and Pollak, 1981), the random effects included in the model are the additive value of the individual (direct effect), the additive value of the mother to produce the

suitable environment (maternal genetic effect) and permanent environmental effects that have an influence on the mother (maternal permanent effect). It is assumed that there is a genetic covariance between direct and maternal effects and there is no correlation between the other effects.

Non Additive Animal Model

The non-additive Animal Model (Henderson, 1985) expands the mixed model equations to include other genetic components such as dominance or epistatic interactions. The procedure developed by Hoeschele and Van Raden (1991) also permits to invert the dominance relationship matrix of large data sets for pedigree files when there is no inbreeding in the population

Marker Assisted Selection

Since the 90´s, molecular information is available due to the advances of techniques of molecular biology. This new source of information gives the opportunity to enhance the response to selection by incorporating it to traditional breeding programs, especially for traits that present difficulties in their improvement by traditional selection. Such traits are those with extremely low heritability (ex. Reproductive traits) and traits whose phenotypes are difficult to obtain (ex. Disease resistance or meat quality).

Following Dekkers (2004), the genetic markers used for selection can be classified into 3 different types, direct markers (or genes), linked markers in population-wide linkage disequilibrium with QTLs (LD markers), and linked markers in population-wide linkage equilibrium with the QTLs (LE markers). Note that a QTL is denoted as a segregating gene affecting the trait of interest. A favorable aspect of direct markers is that are easier to incorporate to an existing breeding program and their potential for increasing the response to selection is greater than the other types of markers. Nevertheless, the difficulty of detecting these genes makes more appealing the use of LD or LE markers. (Dekkers, 2004)

Genomic Selection

Meuwissen et al. (2001) proposed the use of a dense marker map for the prediction of total breeding values and, since then, the procedure is denoted as *Genomic Selection*. The development of commercial genotyping chips made available thousands of single

nucleotide polymorphisms (SNP) markers. It is expected that some of the SNP markers are located near the quantitative trait loci (QTLs) of traits of interest and due to linkage disequilibrium (LD) between them, they should be inherited jointly. In this way, all of QTLs affecting a trait may be in LD with one or more markers. As a consequence, if there are enough markers to cover the whole length of the genome, the additive effects of the QTL can be captured by the markers without the necessity of locating them. Genomic selection procedures consist in estimating the effects associated to the markers in a reference population where genotypes and phenotypes are available for all individuals. These estimates are then used to predict the genetic merit of young individuals with no records available except their genotype. Figure 1 illustrates this genomic selection strategy:

Figure 1. Example of genetic selection.

Reference population

	SNP 1 (A/T)	SNP 2 (C/G)	SNP 3 (T/G)
Animal 1	AA (2)	CG (1)	GG (0)
Animal 2	AT (1)	GG (0)	TT (2)
Animal 3	TT (0)	CC (2)	TG (1)
.....	....	....	....
Animal n	AT (1)	CG (1)	TG (1)

Genomic evaluation using phenotypic records

SNP 1 (A)	SNP 2 (C)	SNP 3 (T)
$\beta_1 = 0.5$	$\beta_2 = -1.5$	$\beta_3 = 2$

Candidates to selection

SNP 1	SNP 2	SNP 3	GBV*

Candidate 1	(AA)	2*0.5	+	(GG)	0*(-1.5)	+	(TG)	1*2 = +3
Candidate 2	(TT)	0*0.5	+	(CC)	2*(-1.5)	+	(GG)	0*2 = -3
Candidate 3	(AT)	1*0.5	+	(CG)	1*(-1.5)	+	(TT)	2*2 = +4.5

*Genomic Breeding Value

From the example above, *n* individuals with phenotypic records are genotyped for 3 SNP markers. The genotypes are codified as 0 and 2 for the two types of homozygous and as 1 for the heterozygous markers. After the genomic evaluation, the estimated effects associated to each marker are obtained. Then, the genotypes of 3 candidates, which are codified in the same way, are multiplied by the estimates of each marker and summed in order to obtain the genomic breeding values. The candidate 3 would be selected for reproduction as the best candidate to selection.

In this example, the marker-effects to be estimated are only 3 and the phenotypic data available for this purpose are much more. Therefore, their estimation presents no difficulty. In reality, thousands of SNP markers are available and their effects need to be estimated from a limited number of phenotypic records. This leads to a situation known as “the large p small n problem”. Traditional methods based on marker regression are unable to deal with this problem unless they introduce some shrinkage on the estimation of the effects. Several methods have been developed to address this problem and to perform genomic selection in practice:

Least squares

This method treats the marker effects as fixed. To deal with the large number of parameters to estimate from few data, a pre-selection of SNPs has to be applied before the analyses. The SNPs with a larger effect are included in the estimation assuming that the rest of the markers do not have any effect. In the original article of Meuwissen et al. (2001), this method gave accuracies of the predicted breeding values of 0.318. The low performance is probably due to the few degrees of freedom available for the estimation of every marker-effect, even after the pre-selection was applied.

Genomic BLUP

G-BLUP is a method similar to the traditional BLUP described by Henderson (1975) that uses a genomic relationship matrix instead of the pedigree relationship matrix. The genomic relationship matrix is built from molecular information in a way that, individuals that share identical by state genotypes for a larger number of markers are more similar and therefore, have larger values in the corresponding cell of the matrix. Misztal et al. (2009) proposed a modification of this method using the pedigree relationship matrix combined with the genomic relationship matrix with weighting parameters between them. The Genomic BLUP methodology does not suffer the large p small n problem since the amount of unknown effects is usually the same as in traditional BLUP (González-Recio et al., 2008).

Bayes A

Originally proposed by Meuwissen et al. (2001), this method assumes a normal distribution on the marker effects with zero mean and variance σ_i^2 associated to each marker. A scaled inversed Chi-squared distribution is assumed for this variance with 4.012 degrees of freedom and scale parameter 0.002. These values were used because they fitted the simulation study of the authors and gave accuracies of 0.8. Since then, they have been applied in many cases without a clear justification.

Bayes B

This method assumes a normal distribution on the marker-effects and variance σ_i^2 associated to each marker just as Bayes A. Bayes B differs from Bayes A when to the assumptions made for the distribution of the variance. A mixture of distributions on the variance is assumed, where the variance is zero with probability π and distributed as in Bayes A with probability 1-π. Although the election of π is arbitrary with no justification, Bayes B provides highly accurate predictions. Meuwissen et al. (2001) obtained accuracies of 0.85 with Bayes B, the highest among the ones obtained using Bayes A, BLUP or least-squares.

Bayes Cπ & Dπ

To address some drawbacks of Bayes A and Bayes B, such as the prior probability of π and the hyper-parameters of the prior distribution of the variance, Habier et al. (2011) described the Bayes Cπ and Bayes Dπ methods. Bayes Cπ method assumes a common variance to all markers with probability 1-π and variance zero with probability π.

Additionally, the proportion π of markers is treated as unknown and is estimated from the data. Bayes Dπ imposes a prior on the scale parameter of the inverse chi-square distribution, which is the prior distribution of the variance of marker-effects. Both methods proved similar to the original methods regarding the accuracies.

Bayesian LASSO

De los Campos et al. (2009) and Usai et al. (2009) proposed the Bayesian LASSO method (Park and Casella, 2008) for genomic selection, where a double exponential prior distribution is assumed for the marker-effects with parameter λ. This method performs a larger shrinkage on the marker-effects than other methods in a way that a large number of markers are estimated with a very small effect, and only a few markers are allowed to have larger effects. The degree of shrinkage is determined by the parameter λ, which has to be estimated previously to the analyses. Park and Casella, (2008) proposed the use of Empirical Bayes by Marginal Maximum Likelihood using an appropriate hyperprior for the estimation of λ. Legarra et al. (2010) proposed a modification of this method (BL2Var) which considers two different variances for the distribution of marker-effects and the residuals. Moreover, there is no need to pre-estimate the parameter λ as it is estimated from the data simultaneously with the marker effects. Up until now Bayesian LASSO has been widely applied for genomic evaluations as it provides accurate predictions for low density genotyping (Usai et al., 2009) and for traits that are regulated by many genes with a small effect (Cleveland et al., 2010).

Elastic Net and SNP pre-selection

Croiseau et al. (2011) proposed the implementation of the elastic net algorithm for genomic selection. This is a combination of Genomic BLUP and Bayesian LASSO weighted by a parameter α which takes values from 0 to 1. When α=0, a BLUP model is defined whereas α=1 a LASSO model is chosen. Additionally, a pre-selection of markers can be applied prior to the analyses. The purpose of this method is to provide a more flexible tool to deal with the large p small n problem.

Non-parametric methods

The need to improve the accuracies of the predictions of breeding values, led to the introduction to the field of genetics of new methodologies from the machine learning

scope. These algorithms are more attractive for application to multiple and more complex situations found in biological systems as they are able to accommodate additive, dominant effects or even epistatic effects. It is believed that these methods can approach the genetic architecture of a trait more than the linear models. Some methods that have been proposed are: Reproducing Kernel Hilbert Spaces Regression (RKHS) (Gianola et al., 2006) which resulted in accuracies similar or even higher than the ones obtained by the Bayesian methods (Gonzalez-Recio et al., 2009), Random Forest (RF), that gives the possibility of capturing interactions between genes and between genes and environment (Sun, 2010) and Neural Networks (NN), which prove to be useful for predicting complex traits as it can capture non-linear relations (Gianola et al., 2011).

Genomic selection for multiple populations

Until now, Genomic selection has been implemented mainly in the dairy cattle industry with encouraging results. The existence of a large enough reference population, due to the world-wide supremacy of the Holstein-Friesian breed, permits us to achieve highly accurate predictions (Hayes et al., 2009). The beef cattle industry though, does not follow the structure of dairy cattle industry and the construction of a large enough reference population presents serious difficulties, due to the existence of many and small populations. Therefore, the evaluation within-breed gives poor results due to the small size of the training sets. Moreover, the estimations obtained from one breed cannot be applied to other breeds as they give very low accuracies (Harris et al., 2008).

To avoid this inconvenience, De Roos et al. (2009) proposed pooling animals from different breeds to obtain a large training set. In his study, two populations that diverged randomly for several generations were simulated. His results showed that adding individuals from the second population to the training set (composed only by the first population), had some effect on the reliability of the genomic breeding values in the first population and it was most beneficial when the heritability of the trait was low. Furthermore, when the two populations had diverged for only few generations and the marker density was high, the information from the second populations was most valuable.

In another study, Hayes et al. (2009) used 3 reference populations, composed by pure Holstein sires, pure Jersey and a population combined between them. They found that training in the admixed population gave similar accuracies for the Holstein as when

training in the pure Holstein population but higher for Jersey. These results show that in situations of small population size, including animals from a different breed may be interesting to increase the accuracy of the predictions.

Further, Kizilkaya et al., (2010) used a multiple population composed by 8 breeds to predict the breeding values for one of the breeds. When the trait of interest had a heritability of 0.5 and the trait was controlled by 50 QTLs, they found accuracies of 0.38. However, when it was controlled by 500 QTLs, they only got an accuracy of 0.22. Their results implies that the predictive ability eroded as the number of QTL increases, and thus, that the genetic architecture of the trait plays an important role on the efficiency of genomic selection for multiple populations.

Finally, Toosi et al. (2010) developed a simulation study that created 4 populations that diverged randomly for 53 generations to set up several types of training sets (admixed and crossbreed). Those training set were used to predict one of the 4 populations. The accuracies obtained were 0.79 when training in purebred, 0.71 when training in admixed that included the validation population and 0.43 when it was not included. These accuracies increased with marker density for all training sets, though with more intensity for the admixed populations. The results of this study proved that marker density also determines the accuracy of genomic selection in multiple populations.

As a general conclusion of these studies, it is known that the predictive ability of the admixed training populations depends on the genetic architecture of the trait and the marker density. It is important to note that the previous simulation studies make the populations diverge exclusively by genetic drift. However, the causes of genetic differentiation of livestock populations include also natural and artificial selection jointly with the genetic drift. Therefore, this study pretends to evaluate the consequences of the number of generations and the cause of reproductive isolation of the populations on the efficiency of genomic selection for multiple populations. In addition, the marker density of genotyping and the number of genes that control the traits will be considered as variables to generate the scenarios of simulation.

OBJECTIVE

The objective of this study is to evaluate the accuracy of the predicted breeding values of purebred populations based on estimates of marker effects in purebred and admixed populations. Moreover, this study pretends to evaluate the influence of the following situations on the accuracy of the predictions and in the persistence of linkage disequilibrium:

1) Causes of reproductive isolation of the populations
2) Number of generations of divergence.
3) Number of genes involved in the genetic determinism of the trait.
4) Marker density of the genotyping.

As a side result, this study also will provide software to simulate populations under different scenarios of genetic evolution.

MATERIAL AND METHODS

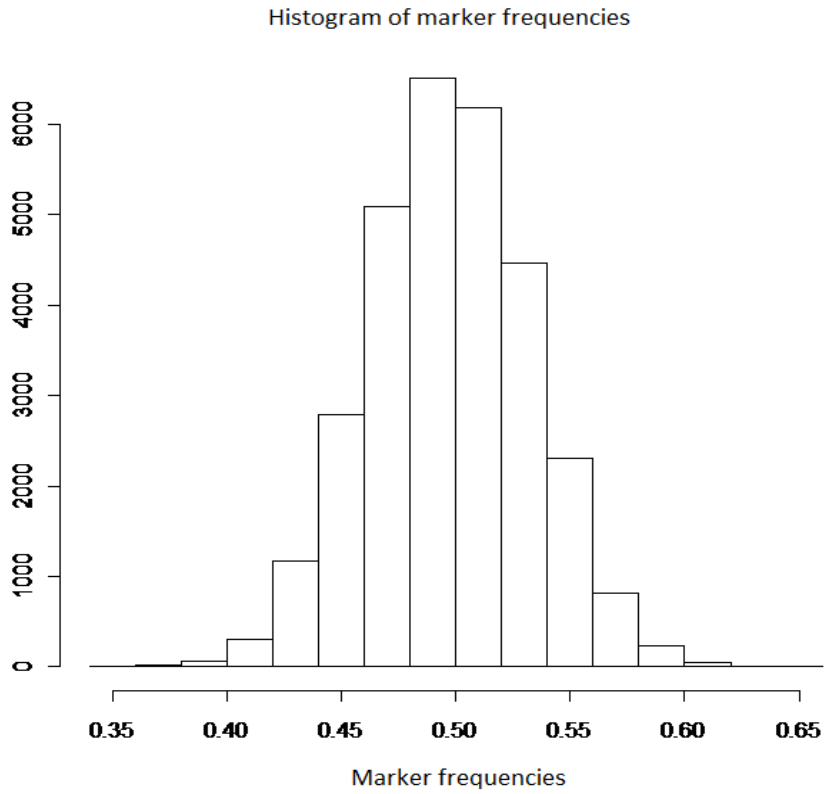
This chapter is structured in three main sections. In first place, the development of the simulation study is described. Secondly, and given the output of the simulation, the Bayesian Lasso method for estimation of markers effects is presented. In third place, the procedures of validation of the genomic prediction and the calculation of the linkage disequilibriumin in single and multiple populations are shown.

Simulation

The simulation study was performed through the development of a Fortran 90 program, which is included in the APPENDIX. The software simulated a base population of 100 unrelated individuals, with sex ratio 1:1. Each individual had 30,000 biallelic markers evenly distributed along the genome. In order to obtain two different scenarios of marker density, a genome of 4 chromosomes and a genome of 20 chromosomes of 1 Morgan each were simulated. Two alleles were considered for each marker and they were coded as 1 and 2. To assign the type of allele to a marker, a random number between 0 and 1 was drawn from a uniform distribution for every allele. The type 1 allele was assigned if the number was lower than 0.5 and the type 2 allele was assigned if the number was greater than 0.5. In this way, the frequencies of the markers in the base population were all near 0.50, as it is presented in Figure 2.

To simulate the next generation, two random numbers, as before, were drawn from a uniform distribution. The first number was used to choose a male and the second one to choose a female. The first marker on every chromosome was used to determine which allele from the pair of chromosomes would be passed on to the next generation. A random number was drawn from a uniform distribution. If it was greater than 0.5 the second allele was passed on and if it was lower the first allele was passed on.

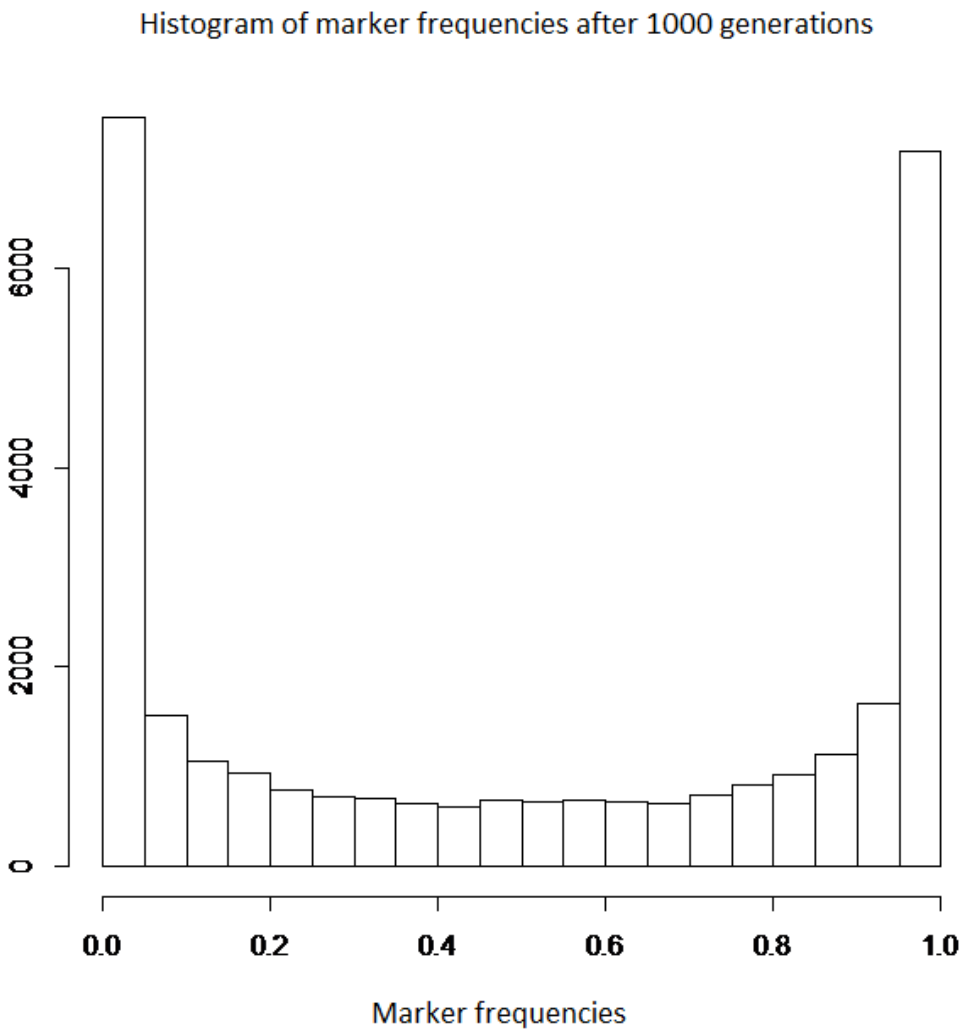
Figure 2. Histogram of marker allelic frequencies in the base population.



To simulate the recombination of the chromosomes, the number of recombination per chromosome was drawn from a Poisson distribution with parameter $\lambda=1$ (1 recombination per chromosome). If there was no recombination, the whole chromosome was passed on as it was. Further, the points of recombination were simulated by the extraction of as many random numbers from a uniform distribution as the number of recombinations. The markers after the point chosen were changed from the first allele to the second allele and vice versa. After the father's chromosome had been recombined and passed on to the son, the mutation was simulated by sampling the number of mutations from a Poisson distribution with parameter $\lambda=1$ and $\lambda=5$ for the 20 and 4 chromosomes scenario, respectively. A total of 1 and 5 mutations per chromosome were simulated in order to have the same mutation rate per locus under all scenarios of marker density. After some tuning, the mutation rate was set at 6.6×10^{-4} per locus with the objective of maintaining the variance after many generations of evolution. Then the points of mutation where chosen by sampling them from a uniform distribution and the allele of the marker selected was changed from type 1 to type 2 or from type 2 to type 1. This procedure was repeated for the chromosomes that the individual received from the

mother. The new generation was then used to create the next one in the same way. The population evolved for 1000 generations, each time using one generation to produce the next. There was no generation overlap and the size of the population was kept at 100 individuals. The allelic frequencies after 1000 generations of evolution are presented in Figure 3.

Figure 3, Histogram of marker allelic frequencies after 1000 generations



At the end of the 1000-generation-evolution, two groups of 100 or 300 markers were selected to simulate the causative mutations of two different traits. To select them, we sampled them from a uniform distribution and they were attributed an effect sampled from a normal distribution with mean 0 and standard deviation 1 in order to have positive and negative effects, many markers with a small effect and a few of them with a large effect. Then, we calculated the frequencies of the causative mutations by

counting the number of type 2 allele in the population and dividing it by the total number of alleles. The additive variance that was created for each trait was calculated as

$$var(\alpha) = \sum_{1}^{m} 2p_i(1 - p_i)\alpha_i^2$$

where p_i is the i^{th} marker frequency, m is the total number of causative mutations and α_i is the substitution effect of the marker i. The heritability for the trait 1 was set 0.2 and for trait 2 was 0.4. The residual variance for each trait was calculated from the equation

$$h^2 = \frac{var(\alpha)}{var(\alpha) + var(r)}$$

At this point 4 subpopulations (A, B, C and D) of 100 individuals were separated from the base population. The subpopulation sA and B were put under a process of phenotypic selection for the traits 1 and 2, respectively. At the same time, subpopulations C and D did not experience any selection and evolved under genetic drift generated by random mating. This strategy allowed simulating 3 different scenarios of time of divergence for the 4 populations that evolved separately for 5, 50 and 200 generations.

To simulate the evolution of the population A, phenotypic values were assigned to all its individuals for the trait 1 as:

$$y_{1k} = \mu + \sum_1^m g_{ik}\, a_i + r_k \,,$$

where y_{1k} is the phenotypic value of individual k for the trait 1, μ is the trait mean which was 100, m is the total number of causative mutations, g_{ik} is the genotype of the individual k at the i^{th}locus, coded as -1, 0 and 1, a_i is the substitution effect of the i^{th} locus and r_k is the residual of individual k sampled from a normal distribution with mean=0 and standard deviation the square root of the residual variance calculated earlier. Then, the phenotypic mean and standard deviation were calculated:

$$mean_p = {\sum_1^k y}/{k}\,, \qquad sd_p = \sqrt{\frac{\sum_1^k (y_k - mean_p)^2}{k-1}}\,,$$

where k is the number of individuals and y_k is the phenotypic value of the individual k. To create the next generation, two individuals with phenotype greater than $(mean_p -$

sd_p), were selected by sampling them with equal probability. This fact implies an intensity of selection of 0.30, approximately. The intensity of selection simulated was low in order to prevent the fixation of the alleles and the shrinkage of the additive variance after many generations of selection. The procedure of allele transmission from one generation to the next was the same as described for the base population. After the various generations of evolution, one more generation was simulated, where the size of the population was increased to 2,000 individuals, creating the breed A. Phenotypic values for the second trait were simulated at this point, the same way as for the first trait but with a trait mean μ=1,000. Finally, breeding values were simulated for every individual, for both traits as:

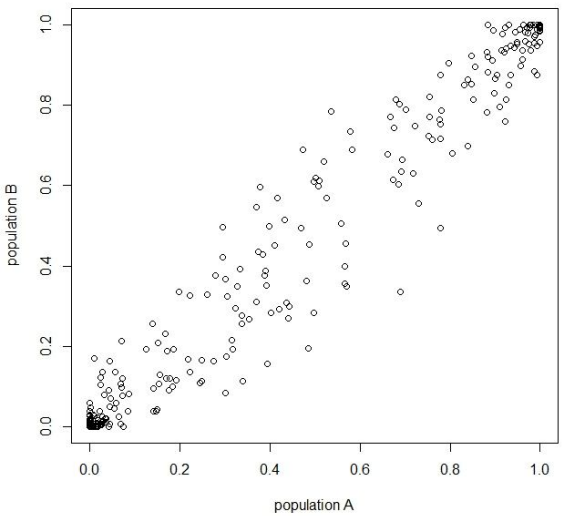
$$tbv = \sum_1^m g_i a_i ,$$

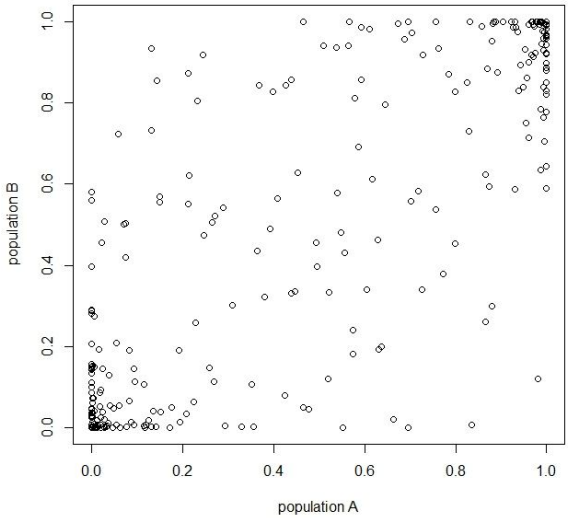
where g_i is the genotype of the i^{th} locus coded as -1, 0- and 1 and a_i is the substitution effect of the causative mutation.

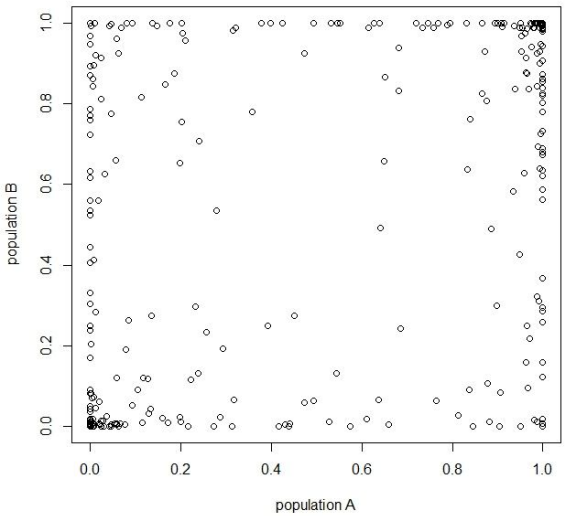
Population B was simulated in the same way as population A except for being selected for the trait 2. Populations C and D were left under random mating. The simulation process for them was the same as the one used for the evolution of the base population. One more generation was added to them as well, to create the breeds B, C and D of 2,000 individuals each. Genotypic records, phenotypic records and breeding values for both traits were obtained at this point from all breeds.

Figure 4, Plots of frequencies of the causative mutations of trait 1 between population A and B after evolving for a) 5 generations, b) 50 generations and c) 200 generations.

a) b) c)







Furthermore, 7 admixed populations of 2,000 individuals were created using the 4 pure breeds. 6 of them were all possible combinations of two pure breeds (A+B, A+C, A+D, B+C, B+D and C+D) with 50% of each breed, while the last one was a combination of all pure breeds (A+B+C+D) with 25% of each breed. A first sampling from a uniform distribution gave us the individual, and a second sampling gave us the breed from which it would come from. The genotype and the phenotypes for both traits of the individual selected were copied to the admixed population.

The purebred and the admixed populations would be used as training sets for estimating marker effects while the 4 pure breeds would be used as validation sets to calculate the accuracy of the predictions. Figure 5 shows the procedure of obtaining the final populations and Table 1 shows the parameters used in the simulation study.

Figure 5. Schematic representation of the simulated population history.

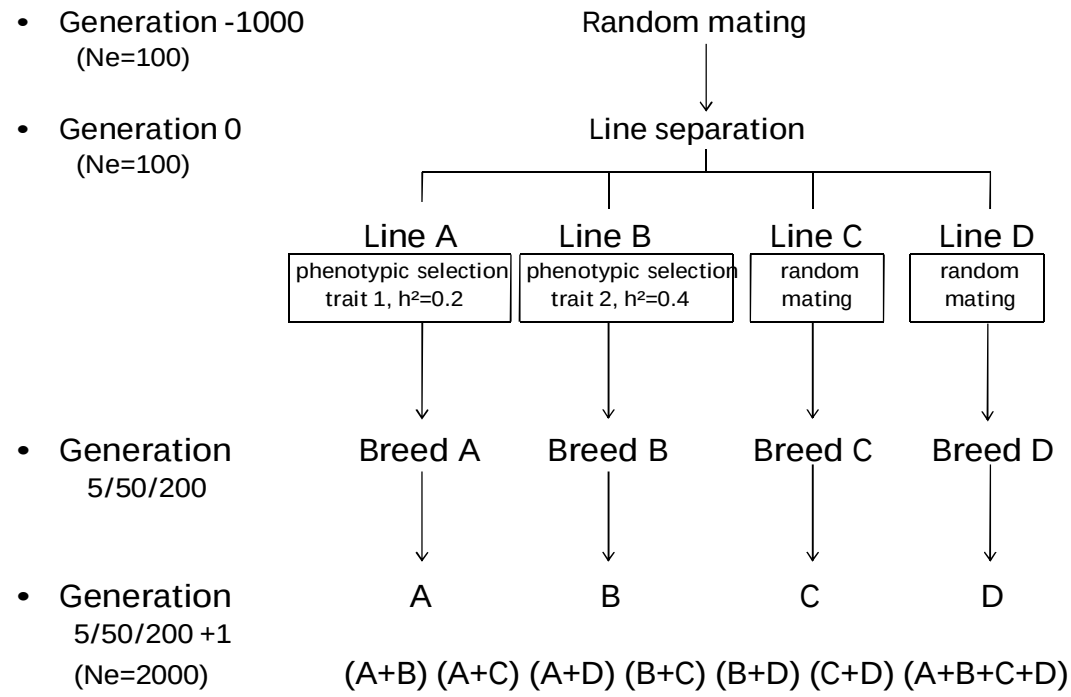


Table 1. The parameters used in the simulation

Genome size	20 M, 4 M
Number chromosomes	20, 4
Number of total markers	30000
Number of causative mutations	100, 300
Marker density per cM	15, 75
Mutation rate per locus	6.6×10^{-4}
Distribution of causative marker effects	Normal (0,1)
Generations of divergence	5, 50, 200
Population size	
Generation -1000 to 0	N = 100
Generation 0 to 5/50/200	N = 100
Generation +1	N = 2000
Heritability	0.2, 0.4

Estimation of marker effects

For each case of simulation, estimation of markers effects was performed with the 4 pure and the 7 admixed populations. The method of choice was the Bayesian Lasso (Park and Casella, 2008) with the approach developed by Legarra et al. (2010). Further, the software GS3 (Legarra et. al., 2012) was used. Among the plethora of procedures to estimate marker effects within the scope of genomic prediction, this procedure was chosen because it presents good prediction results when compared with other methods (Usai et al., 2009; Cleveland et al., 2010), and it is available in the commercial software GS3.

In this study, the model of analysis was:

$$\boldsymbol{y} = \mathbf{1}\mu + \sum \boldsymbol{g}_i a_i + \boldsymbol{e},$$

where $\boldsymbol{y}$ is the vector of phenotypic values of individuals in the training data, μ is a single unknown population mean, $\mathbf{1}$ is a vector of ones, $\mathbf{g}_i$ is a column vector containing the genotypes (coded as 0, 1, or 2) of each individual at locus i, α_i is the random unknown allele substitution effect of marker i, and $\boldsymbol{e}$ is a random vector of unknown residuals with $\boldsymbol{e}|\sigma^2_e \sim \text{MVN}(0, \mathbf{I}\sigma^2_e)$. MVN stands for multi-variable normal distribution.

Legarra's et. al. (2010) modification of the Tibshirani's (1996) original Lasso considers two different variances, one for the conditional distribution of marker effects, and another for the residuals. This is more similar to the classical models used in quantitative genetics. The distribution of the marker effects is a conditional Laplace distribution and it can be written as:

$$\boldsymbol{g}|\lambda, \sigma_{\alpha}^{2} \sim \prod_i \frac{\lambda}{2\sigma_\alpha} \exp \frac{-\lambda|\alpha_i|}{\sigma_\alpha}, \qquad \boldsymbol{e}|\sigma^2_e \sim \text{MVN}(0, \mathbf{I}\sigma^2_e),$$

which is equivalent to the original form of Tibshirani's Lasso

$$\boldsymbol{g}|\lambda \sim \prod_i \frac{\lambda}{2} \exp(-\lambda|\alpha_i|), \qquad \boldsymbol{e}|\sigma^2_e \sim \text{MVN}(0, \mathbf{I}\sigma^2_e),$$

because only the ratio λ/σ_α is used and it cannot be estimated separately. Also, λ determines the variance of the marker effects using

$$Var(\alpha) = 2/\lambda^2 .$$

With this modification it's not needed for the λ to be estimated previously as it's estimated simultaneously with the markers effects.

In this study the prior distribution for the residual variance and the genetic variance was assumed to be uniform. The prior for λ was assumed uniform as well, bound between 0 and 1,000,000.

For estimation purposes, the markers selected as causative mutations for the two traits, were excluded from the marker panel, and only the neutral markers were used for the estimation. A Markov chain Monte-Carlo (MCMC) of length 100,000 cycles with a burn-in period of 20,000 cycles was conducted.

Validation of genomic prediction and calculation of linkage disequilibrium

Once estimates of marker effects were obtained from the training sets, the estimated breeding values of individual k ($GEBV_k$) in the validation data set (A, B, C and D purebred populations) were computed as

$$GEBV_k = \sum_{i=1}^{m} g_{ik} \hat{a}_i ,$$

where g_{ik} is the genotype (-1, 0, and 1) of individual k at locus i, $\hat{a}_i$ is the estimated effect of marker i, and m is the total number of markers. Accuracy was calculated as the

raw correlation between the estimated and the simulated breeding values of individuals in the 4 validation sets. This accuracy was used to compare the performance of the different scenarios and training sets simulated. All scenarios were replicated 10 times, and the results were averaged across replicates.

To evaluate the extent and the magnitude of linkage disequilibrium in the training populations and its impact on the accuracy of genomic predictions, linkage disequilibrium between the neutral markers and the causative mutations was calculated. A small Fortran 90 program was developed to calculate the observed gametic frequencies f_{11}, f_{22}, f_{12} and f_{21} of marker-causative mutation pairs and then the linkage disequilibrium was calculated as described by Falconer, using

$$D = (f_{11} * f_{22}) - (f_{12} * f_{21}).$$

Further, to evaluate the persistence of LD phase across validation and training populations, the marker-causative mutation pairs with D>0.1 and D<-0.1 in the validating populations were chosen and their D in all training population was compared. The comparison was calculated as the percentage of the pairs in strong LD in the validation populations that appeared also in strong LD in the training populations.

Additionally to the D, the correlation coefficient between pairs of loci was calculated. The correlation coefficient is a measure of LD, an alternative to D, expressed as:

$$r = \frac{D}{\sqrt{p(1-p)q(1-q)}}$$

and its advantage is that it is adjusted to the loci having different allele frequencies. Its permits to distinguish better the higher values of LD and serves as a comparable measure of LD between populations with different allele frequencies.

RESULTS AND DISCUSSION

In this chapter, the results of this study are presented and discussed. The objectives were to compare the predictive ability of several purebred and admixed populations under different scenarios of marker density, number of causative mutations affecting the traits, number of generations of divergence and type of reproductive isolation of the populations. Initially, the scenario with 15 markers per cM and 300 causative mutations for each trait is examined. In first place, the predictive ability of the purebred populations is evaluated. Secondly, the accuracies obtained from admixed populations composed of 2 purebred are compared and finally, the performance of the admixed population composed of all 4 purebred is presented. After that, these results are compared in terms of the accuracies of predicted breeding values with the ones from the two other scenarios. In one case, when there are 75 markers per cM, and, in the other, when only 100 causative mutations per trait are simulated.

For the first case, a population of 100 individuals with 30,000 biallelic markers each, evenly distributed along 20 chromosomes of 1M was simulated. From these markers, two groups of 300 markers were randomly chosen in order to simulate the causative mutations of two traits with heritabilities of 0.2 and 0.4. The population mated randomly for 1,000 generations. At generation 0, 4 new populations of 100 individuals were created from the base population. The first was put under selection for the trait 1, and the second for the trait 2, whereas the last two populations were left to evolve under random mating. After 5, 50 and 200 generations of evolution, the final purebred populations (A, B, C and D) were created by increasing their size to 2,000 individuals each. Moreover, several admixed populations were created combining phenotypic and genotypic records from the 4 purebred populations. There were created 6 admixed populations combining 2 purebred with 50% from each, and one admixed combining all 4 purebreds with 25% from each. The phenotypic and genotypic records of all purebred and admixed population were used for the genomic evaluation while the 4 purebred served as validation sets, as it can be seen in Figure 5 in the previous chapter.

Purebred populations

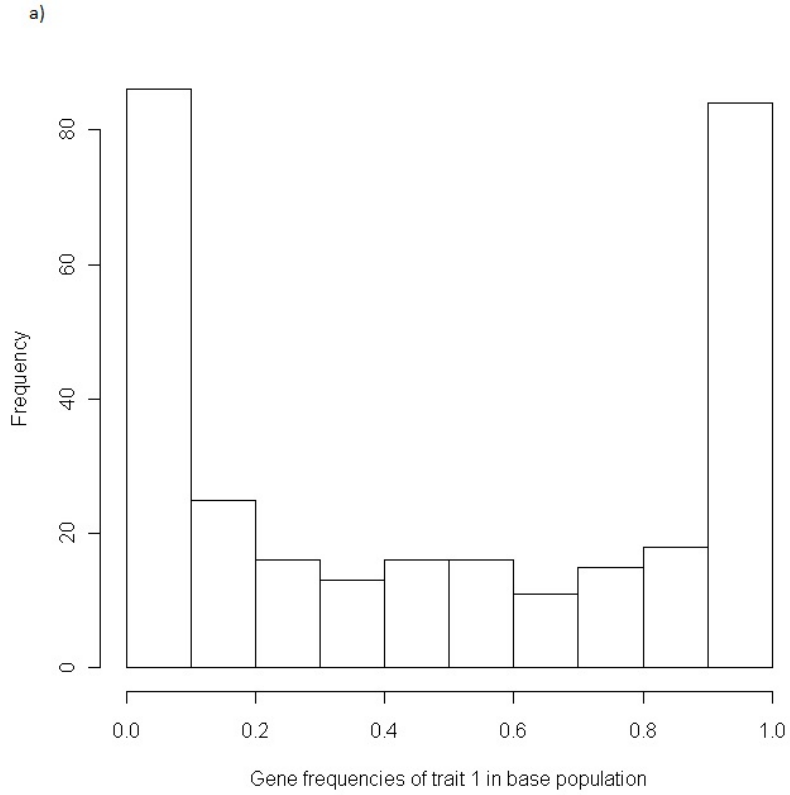
Table 2 shows the accuracies of the predicted breeding values for the 4 validation sets (A, B, C and D) for both traits, obtained from training in the same 4 populations. The upper half of the table shows the evaluations for the first trait, while the lower half for

the second trait. The validation sets are located in the first column and the training sets are found in the line above the results. The table is separated in 3 parts which show the results obtained for the same populations, under 3 scenarios of number of generations of divergence (5, 50 and 200 generations). The diagonal elements of every sub-table show the accuracies obtained when training and predicting in the same population, while the elements outside the diagonal show the predictive ability of the populations over the others.

The accuracies achieved overall, ranged between -0.023 (Trait 1, prediction in population B and validation in population A with 200 generations of divergence) and 0.822 (Trait 2, prediction and validation in population C) with the accuracies for the second trait being generally higher than the ones for the first trait. This difference in the accuracies between the two traits results from their different heritabilities. As expected, traits with higher heritability give better predictions (Falconer and Mackey, 1996).

When training and predicting in the same population, the accuracies obtained were around 0.7 for trait 1 and 0.8 for trait 2 and were maintained at the same level in all 3 scenarios of generations of evolution. Exceptions to this were populations A and B. Population A showed a decline to its predictive ability for trait 1 of 4.7% and 15.1% after 50 and 200 generations of evolution respectively. Likewise, the accuracy of the predictions for the trait 2 made using population B declined 2.2% and 10.4% for 50 and 200 generations. The loss of the accuracy observed is produced due to the fact that these populations were put under selection for the traits of interest (population A selected for trait 1 and population B selected for trait 2). The process of selection tends to change the allele frequencies of the causative mutations and force them near fixation or loss, as it is shown in Figure 6, for one specific replicate. As a result, there is a loss of genetic variance, as it is illustrated in Figure 7, which is essential for estimating marker effects and predicting breeding values (Falconer and MacKay, 1996). In addition, a slight decrease is also observed for populations A and B for the trait that is not used for selection, because of the lower effective size caused by the selection process.

Figure 6. The histogram of the gene frequencies of trait 1 in a) in the base population and b) after 200 generations of selection.



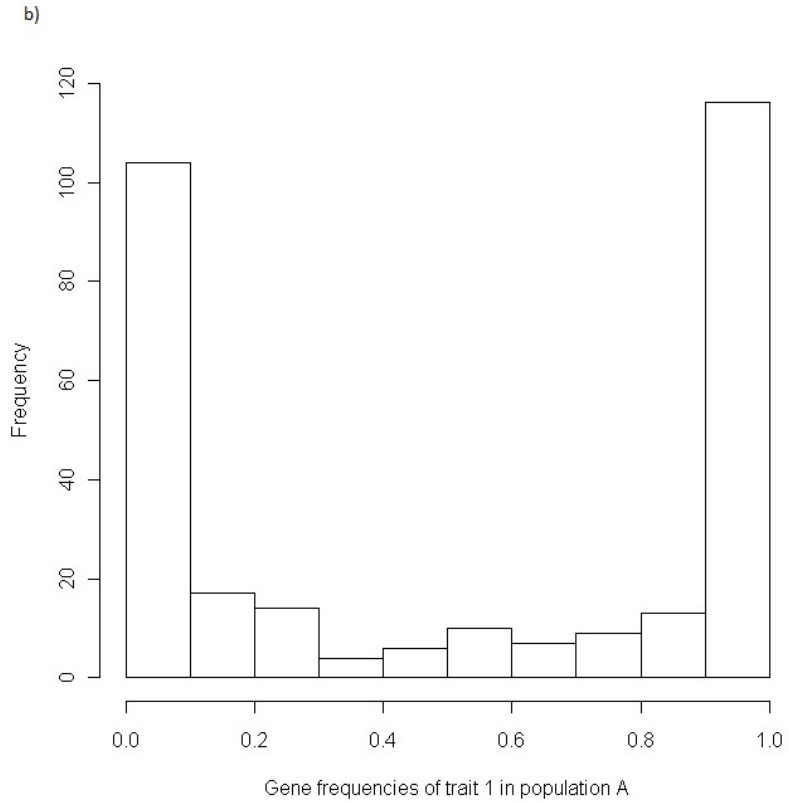
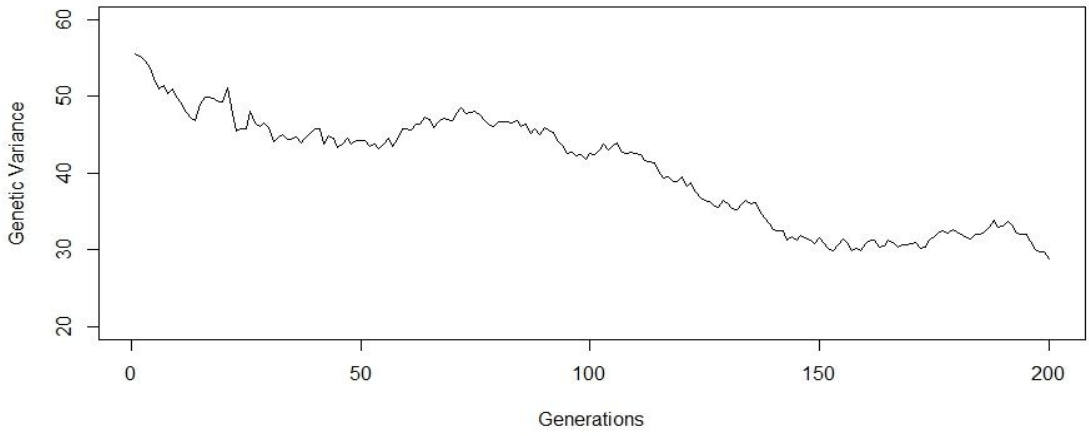


Figure 7. Graphical representation of the loss of genetic variance in 200 generations of selection.



The results show that the predictive ability of one population over the others is low to be used for selection purposes, as it is even lower than the one obtained by mass selection (0.44 for trait 1 and 0.63 for trait 2, respectively). When the populations are separated for 5 generations the accuracies ranged around 0.3 and 0.4 for train 1 and trait 2 respectively, which declined rapidly with the number of generations of separation, being 0.1 and 0.14 after 50 generations and very close to 0 after 200 generations. This occurs because the forces of selection and genetic drift that act, make that the alleles of the genes are fixed or lost in a different way in every population, and therefore the linkage disequilibrium extent is very different in every population, as it is presented in Figure 4 in the Material and Methods section.

Besides the accuracies, the linkage disequilibrium (LD) between the causative mutations of both traits together and all the markers was calculated using

$$D = (f_{11} * f_{22}) - (f_{12} * f_{21})$$

as described by Falconer and Mackay (1996). The accuracy of the predictions depends largely on the extent of the LD between the markers and the genes. When more markers are in LD with a gene and when the LD is strong, then the accuracy improves. Likewise, when two populations share the same markers, genes and LD patterns then more accurate predictions are expected.

Firstly, LD was calculated separately in all populations. Afterwards, the causative mutations-marker pairs that presented a LD with a *D* higher than 0.1, in the 4 validation sets (A, B, C and D), were selected. These pairs were then looked up in all populations to see how many of them presented also a *D* higher than 0.1 as in the validation sets and the percentage of common pairs in high LD was calculated.

Table 3 presents the percentage of common pairs in high LD between the validation sets and the training sets under the 3 scenarios of number of generations of divergence. Beneath the percentage of common pairs, the correlation between the LD in the validation set and the LD in the training set of the selected gene-marker pairs is also shown. Like in table 1, the validation sets are located in the first column and the training sets in the line above the results. The elements of the diagonal are all 100% (and 1 for the correlation) because the same population is used as training and validation set.

The gene-marker pairs in high LD that the training sets share with the validation sets reach up to 49%-56% when the populations are separated for 5 generations and drops to 11%-15% after 50 generations and to 2%-4% when they are separated for 200 generations. Likewise, the correlation of the LD between populations is higher than 0.9 in the case of 5 generations of divergence. But drops to half after 50 generations and reaches a value close to 0 after 200 generations. In all cases, the populations that were put under selection (A and B) present the lowest values of percentage and correlation. This occurs because the force of selection acts additionally to the force of genetic drift and creates patterns of LD specific of the population (Falconer and MacKay, 1996).

These results of the LD extent are consistent with the results of the predictive ability of each population. As it can be noticed, when the populations are separated for 5 generations they share around 50% of gene-marker pairs in high LD and the accuracies obtained are around 0.3 (for trait 1), half of the accuracy obtained from evaluating and predicting in the same population. Moreover, the predictive ability of one population over the other declines with the generations of divergence with the same rate as the percentage of gene-marker pairs in high LD that they share.

Table 2. Accuracy (standard error) of the predicted breeding values of the 4 purebred populations obtained from training in the same populations.

		5 Generations				50 Generations				200 Generations			
	Tr* / **Val***	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
Trait 1 (h^2=0.2)	**A**	0.702 (0.008)	0.304 (0.015)	0.313 (0.019)	0.261 (0.024)	0.669 (0.007)	0.076 (0.016)	0.107 (0.031)	0.076 (0.022)	0.596 (0.015)	-0.023 (0.019)	0.005 (0.026)	0.034 (0.021)
	B	0.270 (0.022)	0.718 (0.011)	0.298 (0.023)	0.286 (0.023)	0.047 (0.026)	0.716 (0.011)	0.119 (0.024)	0.117 (0.022)	0.032 (0.019)	0.706 (0.008)	-0.017 (0.026)	-0.009 (0.021)
	C	0.305 (0.13)	0.276 (0.026)	0.713 (0.009)	0.299 (0.017)	0.087 (0.026)	0.105 (0.021)	0.689 (0.010)	0.103 (0.021)	0.025 (0.025)	0.016 (0.016)	0.703 (0.014)	0.010 (0.023)
	D	0.298 (0.026)	0.356 (0.018)	0.283 (0.017)	0.705 (0.012)	0.074 (0.030)	0.093 (0.024)	0.091 (0.019)	0.709 (0.010)	0.030 (0.018)	-0.002 (0.023)	0.009 (0.019)	0.691 (0.017)
	Tr / **Val**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
Trait 2 (h^2=0.4)	**A**	0.808 (0.009)	0.403 (0.026)	0.413 (0.027)	0.379 (0.022)	0.812 (0.007)	0.072 (0.023)	0.145 (0.028)	0.138 (0.027)	0.789 (0.009)	-0.004 (0.025)	0.044 (0.020)	0.073 (0.026)
	B	0.355 (0.014)	0.786 (0.005)	0.361 (0.018)	0.388 (0.013)	0.084 (0.011)	0.769 (0.011)	0.095 (0.017)	0.089 (0.018)	-0.005 (0.032)	0.704 (0.017)	0.012 (0.019)	-0.001 (0.018)
	C	0.406 (0.013)	0.428 (0.019)	0.822 (0.006)	0.429 (0.018)	0.126 (0.032)	0.074 (0.018)	0.816 (0.008)	0.128 (0.030)	-0.013 (0.027)	-0.009 (0.018)	0.813 (0.007)	-0.018 (0.018)
	D	0.368 (0.016)	0.384 (0.033)	0.405 (0.020)	0.809 (0.007)	0.129 (0.021)	0.075 (0.025)	0.137 (0.017)	0.795 (0.008)	0.020 (0.021)	-0.004 (0.022)	0.031 (0.017)	0.806 (0.006)

*Tr = Training sets, *Val = Validation sets

Table 3. Percentage (standard error) of gene-marker pairs in high LD in the validation sets found in the training sets, and correlation (standard error) between the LD of these pairs in the validation sets and the other purebred populations.

	5 Generations				50 Generations				200 Generations			
Tr* / Val*	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
A	100% (0.849)	49% (0.923)	49.8% (1.071)	49%	100% (0.242)	11.5% (0.589)	11.9% (0.371)	12.3%	100% (0.234)	2.2% (0.175)	3.4% (0.207)	3.3%
	1	0.917 (0.002)	0.926 (0.002)	0.923 (0.002)	1	0.348 (0.016)	0.375 (0.010)	0.381 (0.010)	1	0 (0.007)	0.020 (0.013)	0.025 (0.006)
B	48.8% (1.015)	100%	50.7% (1.032)	49.7% (1.070)	11.3% (0.473)	100%	12.4% (0.726)	13.1% (0.533)	2.6% (0.221)	100%	3.7% (0.312)	3.5% (0.241)
	0.917 (0.003)	1	0.923 (0.003)	0.925 (0.001)	0.353 (0.011)	1	0.376 (0.012)	0.386 (0.012)	-0.005 (0.007)	1	0.031 (0.012)	0.031 (0.007)
C	53.6% (0.746)	54.8% (0.567)	100%	55.1% (0.621)	13.4% (0.346)	14.2% (0.317)	100%	14.7% (0.676)	3% (0.192)	2.8% (0.329)	100%	3.9% (0.362)
	0.932 (0.002)	0.932 (0.002)	1	0.940 (0.001)	0.390 (0.008)	0.399 (0.009)	1	0.416 (0.007)	0.011 (0.010)	0.020 (0.006)	1	0.036 (0.008)
D	53.5% (0.806)	54.5% (0.700)	55.9% (0.854)	100%	13.6% (0.279)	14.9% (0.612)	14.3% (0.699)	100%	3% (0.136)	2.7% (0.199)	4% (0.294)	100%
	0.931 (0.002)	0.931 (0.001)	0.942 (0.002)	1	0.400 (0.010)	0.404 (0.010)	0.423 (0.009)	1	0.029 (0.008)	0.026 (0.004)	0.037 (0.007)	1

*Tr = Training sets, *Val = Validation sets

Admixed Populations x2

The use of admixed populations for genomic evaluation was first proposed by de Roos et al. (2009) with the purpose of overcoming the obstacles that genomic selection is facing when to its implementation in the beef cattle industry. These obstacles are the large number of populations and their reduced population size, due to which the within-breed evaluation is expected to give very low accuracies. Moreover, the estimations made in one population do not produce accurate predictions for a different population (Harris et al., 2008). The use of phenotypic and genotypic data from different populations permits the construction of a reference population large enough as to increase the accuracy of the predictions that can be applied in all the populations involved. Moreover, the larger variability of this population may contribute further in the increase of the accuracy (Falconer and MacKay, 1996).

In this study, the admixed populations were simulated combining data of individuals from the 4 purebred populations. For this purpose, 1,000 individuals were chosen randomly from each one of the 4 purebred populations. Afterwards, these data were combined in pairs of two creating 6 admixed populations (A+B, A+C, A+D, B+C, B+D and C+D) of 2,000 individuals each with 50% from each purebred. These admixed datasets were used to predict the breeding values of the 4 purebred populations.

Table 4, structured as before, shows the accuracies of the predicted breeding values of the validation sets obtained from training in 4 admixed populations (A+B, A+C, B+D and C+D). The results from the A+D and B+C populations are not shown as they are similar to the A+C and B+D, respectively.

The highest accuracies for each trait were obtained for the populations that were included in the admixed training sets and when the populations were separated just for 5 generations. These values ranged around 0.35 and 0.5 for traits 1 and 2, respectively. As the number of generations increased, the accuracy of the predictions for the populations under selection declined. After 50 and 200 generations, Population A lost 11% (0.313 and 0.300) and 28% of accuracy (0.257 and 0.253) for trait 1 and population B lost 8% (0.433 and 0.441) and 27% (0.357 and 0.353) of accuracy for trait 2. Meanwhile, in the cases where selection had not acted the accuracies were maintained almost at the same levels.

The predictive ability of the admixed populations over the ones that did not participate in them was extremely low even when the populations are separated just for 5 generations. The values obtained were between 0.11 and 0.17 for the first trait and between 0.17 and 0.24 for the second trait and declined rapidly to 0 after 50 generations of divergence.

Regarding the linkage disequilibrium (Table 5), the admixed populations presented 65 to 70% of the gene-marker pairs in high LD found in the validations sets that were included in the mixture. The correlation of the LD of these pairs between the validation sets and the admixed populations was 0.98. This percentage declined to around 31% when the populations were separated for 50 generations, and to around 25% in the case of 200 generations. Nevertheless, the correlations between the LD of the pairs of markers did not decline in the same rate. They were maintained above 0.9 in the case of 50 generations and dropped at 0.85 after 200 generations. Although the percentage of common gene-marker pairs in high LD was reduced with the generations of divergence, the accuracies remained at the same levels showing a greater resemblance to the evolution of the correlations between the LD of the same pairs in the different populations throughout the generations.

When comparing the LD between the admixed and the purebred populations that were not included in the mixture, it can be noticed that the admixed populations present 49% to 56% of the pairs found in the purebred populations when they are separated for 5 generations and this percentage is reduced to 11-15% and 7-11% after 50 and 200 generations respectively. Similarly, the correlation between the LD that these pairs present in every population, starts from 0.94 and declines rapidly to 0.5 and finally to 0 with the generations. Though the admixed populations share the 50% of the pairs in high LD with the purebred populations that are not included in the mixture, and the correlation between the LD of these pairs is extremely high, the accuracies obtained are very low.

Table 4. Accuracy (standard error) of the predicted breeding values for the 4 purebred populations obtained from training in 4 admixed populations (A+B, A+C, B+D and C+D).

		5 Generations				50 Generations				200 Generations			
	Tr* / **Val***	**A+B**	**A+C**	**B+D**	**C+D**	**A+B**	**A+C**	**B+D**	**C+D**	**A+B**	**A+C**	**B+D**	**C+D**
Trait 1 (h^2=0.2)	**A**	0.360 (0.007)	0.335 (0.009)	0.107 (0.018)	0.128 (0.020)	0.313 (0.011)	0.300 (0.015)	0.015 (0.023)	0.074 (0.018)	0.257 (0.012)	0.253 (0.006)	-0.010 (0.018)	0.025 (0.013)
	B	0.370 (0.011)	0.113 (0.016)	0.352 (0.008)	0.131 (0.025)	0.331 (0.010)	0.008 (0.031)	0.350 (0.009)	0.072 (0.015)	0.332 (0.010)	0.001 (0.024)	0.335 (0.016)	-0.016 (0.021)
	C	0.125 (0.023)	0.353 (0.009)	0.110 (0.020)	0.348 (0.012)	-0.014 (0.019)	0.323 (0.010)	0.010 (0.022)	0.344 (0.011)	0.015 (0.025)	0.344 (0.011)	0.032 (0.017)	0.352 (0.013)
	D	0.157 (0.028)	0.140 (0.021)	0.353 (0.009)	0.348 (0.009)	0.039 (0.018)	0 (0.022)	0.332 (0.012)	0.340 (0.011)	0.017 (0.023)	0.003 (0.020)	0.349 (0.011)	0.324 (0.019)
	Tr / **Val**	**A+B**	**A+C**	**B+D**	**C+D**	**A+B**	**A+C**	**B+D**	**C+D**	**A+B**	**A+C**	**B+D**	**C+D**
Trait 2 (h^2=0.4)	**A**	0.520 (0.015)	0.502 (0.013)	0.193 (0.022)	0.225 (0.028)	0.504 (0.010)	0.499 (0.016)	0.068 (0.022)	0.076 (0.020)	0.473 (0.011)	0.465 (0.012)	-0.002 (0.027)	-0.002 (0.017)
	B	0.481 (0.010)	0.209 (0.020)	0.477 (0.011)	0.178 (0.019)	0.443 (0.017)	0.047 (0.019)	0.441 (0.014)	0.051 (0.020)	0.357 (0.017)	-0.007 (0.017)	0.353 (0.019)	0.034 (0.016)
	C	0.228 (0.015)	0.518 (0.014)	0.245 (0.017)	0.526 (0.014)	0.048 (0.024)	0.513 (0.014)	0.025 (0.014)	0.520 (0.010)	-0.009 (0.021)	0.504 (0.008)	-0.022 (0.020)	0.514 (0.008)
	D	0.167 (0.019)	0.189 (0.020)	0.521 (0.007)	0.517 (0.011)	0.009 (0.021)	0.054 (0.028)	0.506 (0.008)	0.487 (0.007)	-0.009 (0.028)	0.022 (0.021)	0.498 (0.017)	0.494 (0.015)

*Tr = Training sets, *Val = Validation sets

Table 5. Percentage (standard error) of gene-marker pairs in high LD in the validation sets found in the training sets and correlation (standard error) between the LD of these pairs in the validation sets and the admixed populations.

	5 Generations				50 Generations				200 Generations			
Tr* / Val*	**A+B**	**A+C**	**B+D**	**C+D**	**A+B**	**A+C**	**B+D**	**C+D**	**A+B**	**A+C**	**B+D**	**C+D**
A	65.6% (0.869)	65.6% (0.627)	49.8% (0.913)	49.5% (1.039)	31.8% (0.553)	30.4% (0.828)	11.7% (0.476)	11.2% (0.630)	27% (0.764)	24.9% (0.405)	8.2% (0.347)	6.7% (0.257)
	0.982 (0.001)	0.984 (0.001)	0.937 (0.002)	0.939 (0.002)	0.905 (0.003)	0.915 (0.002)	0.436 (0.009)	0.447 (0.011)	0.842 (0.004)	0.863 (0.004)	0.035 (0.008)	0.033 (0.010)
B	65.6% (0.905)	50.6% (0.966)	66.3% (0.731)	50.7% (1.027)	31.6% (0.368)	11.6% (0.690)	31.3% (0.954)	11.7% (0.648)	26.1% (0.561)	8% (0.353)	25.1% (0.796)	6.4% (0.346)
	0.982 (0.001)	0.936 (0.002)	0.984 (0.001)	0.938 (0.002)	0.905 (0.002)	0.441 (0.013)	0.910 (0.002)	0.453 (0.010)	0.838 (0.004)	0.027 (0.009)	0.857 (0.003)	0.038 (0.011)
C	55.6% (0.666)	68.8% (0.714)	55.8% (0.496)	70% (0.536)	15.2% (0.540)	31.7% (0.617)	14.8% (0.502)	32.3% (0.655)	10.7% (0.492)	26.1% (0.390)	9% (0.344)	25.3% (0.672)
	0.949 (0.001)	0.985 (0.001)	0.952 (0.001)	0.987 (0.001)	0.485 (0.010)	0.911 (0.002)	0.492 (0.007)	0.919 (0.002)	0.042 (0.008)	0.844 (0.003)	0.040 (0.009)	0.866 (0.002)
D	55.8% (0.781)	55.4% (0.807)	69.4% (0.484)	70.2% (0.684)	15.8% (0.583)	13.8% (0.564)	33.4% (1.028)	31.2% (0.804)	11% (0.437)	8.6% (0.311)	27.7% (0.538)	24.3% (0.496)
	0.949 (0.001)	0.952 (0.001)	0.984 (0.001)	0.987 (0.001)	0.499 (0.008)	0.499 (0.009)	0.913 (0.001)	0.918 (0.002)	0.058 (0.007)	0.046 (0.006)	0.850 (0.002)	0.861 (0.003)

*Tr = Training sets, *Val = Validation sets

Admixed Population x4

Besides the admixed populations composed by two purebred, one additional admixed population was created. In this mixture, all 4 purebred populations contributed with data from 500 individuals each, chosen randomly. Table 6 shows the accuracies of the estimated breeding values when the A+B+C+D population was used for training.

The highest values were obtained in the case of 5 generations of separation and ranged around 0.52 for all the validation sets for trait 1 and around 0.60 for trait 2. The values for trait 1 declined approximately 20% for all validation sets after 50 generations and an additional 25% after 200 generations. For the first trait, the highest reduction of accuracy (36%) was presented in population A, which was selected for this characteristic. The loss of accuracy for the second trait was about 30% after 50 generations. The accuracy for the population B presented an additional reduction of 20% after 200 generations due to the fact that this population was selected for this trait. Yet, the reduction of the accuracies for the rest of the populations was much smaller ranging from 1% (for populations C and D) to 10% (for population A).

The admixed population shares similar percentages of gene-marker pairs in high LD with all the purebred populations (Table 7) because all purebred contributed with the same proportion in the mixture. In the case of 5 generations of divergence the admixed population has around 62% of the pairs in high LD found in all 4 purebred populations. However, it should be noted this figure is slight lower (56.8 and 57.6) for populations A and B. This percentage decreases rapidly to 20% and 9% after 50 and 200 generations respectively. As before, it should be noted that populations A and B presented smaller values (16 and 8%). As in the admixed populations composed by two purebred, the correlation of the LD of these pairs in the admixed and the purebred populations does not decline at the same ratio as the percentages. It starts at 0.97 in the first case, 0.82 in the second and finally 0.69 after 200 generations.

Table 6. Accuracy (standard error) of the predicted breeding values for the 4 purebred populations obtained from training in the admixed population (A+B+C+D).

		5 Generations	50 Generations	200 Generations
Trait 1 (h^2=0.2)	**Tr* / Val***	**A+B+C+D**	**A+B+C+D**	**A+B+C+D**
	A	0.510 (0.011)	0.386 (0.018)	0.245 (0.014)
	B	0.517 (0.018)	0.444 (0.031)	0.326 (0.028)
	C	0.532 (0.015)	0.416 (0.021)	0.294 (0.032)
	D	0.523 (0.015)	0.421 (0.033)	0.305 (0.033)
Trait 2 (h^2=0.4)	**Tr / Val**	**A+B+C+D**	**A+B+C+D**	**A+B+C+D**
	A	0.598 (0.028)	0.428 (0.015)	0.383 (0.013)
	B	0.535 (0.033)	0.358 (0.016)	0.286 (0.019)
	C	0.612 (0.033)	0.427 (0.016)	0.408 (0.017)
	D	0.587 (0.034)	0.408 (0.009)	0.405 (0.013)

Table 7. Percentage (standard error) of the gene-marker pairs in high LD in the validation sets found in the training sets, and correlation (standard error) between the LD of these pairs in the validation sets and the admixed population.

	5 Generations	50 Generations	200 Generations
Val \ Tr	**A+B+C+D**	**A+B+C+D**	**A+B+C+D**
A	56.8% (0.875)	16.3% (0.598)	7.9% (0.172)
	0.971 (0.001)	0.814 (0.003)	0.690 (0.005)
B	57.6% (0.958)	16.8% (0.753)	7.8% (0.279)
	0.970 (0.001)	0.813 (0.005)	0.688 (0.010)
C	62.7% (0.588)	20% (0.434)	9.1% (0.360)
	0.977 (0.001)	0.824 (0.004)	0.662 (0.009)
D	62.9% (0.761)	20.2% (0.763)	9.6% (0.305)
	0.978 (0.001)	0.831 (0.002)	0.672 (0.007)

From the results presented above, it can be seen that the purebred populations performed better under all scenarios, as previously reported by Kizilkaya et al. (2010) and Toosi et al. (2010). Though, the admixed populations presented some results that were not so clearly expected. The population A+B+C+D gave similar accuracies to the ones obtained from the other admixed x2 populations for the purebred populations involved and, in some cases, even higher. Yet, the LD extent that this population presents does not match with the pattern of the results of accuracy.

One example of this discrepancy is presented with more detail in Table 8. It shows the accuracies of the predicted breeding values for the first trait for population A obtained from training in this purebred population and two admixed populations, A+B and A+B+C+D, under the scenario of 50 generations of divergence. The percentage of gene-marker pairs in high LD that each training set shares with the population A, together with the correlation between the LD values that these pairs present in the validation set and the training sets (between brackets) are shown beneath the correspondent accuracies. The population A+B maintains the 31.8% of the gene-marker pairs in high LD found in population A and yields an accuracy of 0.313. On the contrary, population A+B+C+D gave an accuracy of 0.386 while presenting just 16.3% of these pairs. Thus, the accuracies obtained from these two training sets are apparently not consistent with the results from the LD.

Table 8. Accuracy of predicted breeding values and percentage (correlation) of gene-marker pairs in high LD for population A when training in populations A, A+B and A+B+C+D.

		A	**A+B**	**A+B+C+D**
A	Accuracy	0.669	0.313	0.386
	LD (corr.)	100% (1)	31.8% (0.905)	16.3% (0.814)

To investigate the causality of this phenomenon, the populations from the example above were taken and examined thoroughly. Firstly, the estimates of the marker effects obtained from these populations were represented graphically (figure 8). The magnitude of the estimated markers effects from population A+B+C+D ranged between -0.10 and +0.10 units and they were more similar to the estimates from the population A, which ranged from -0.03 to 0.03. On the contrary, the estimations from population A+B were

extremely high, ranging between -2 and +2. Similar results were observed in other replicates for the admixed x2 populations.

Afterwards, in order to examine the extent of LD in all these populations, the correlation coefficient (r) between pairs of loci was calculated. The average r of two consecutive loci, of loci that are located in the same chromosome, and of loci that are located in different chromosomes was calculated for all three populations and is presented in the Table 9. The average r of two consecutive loci in the population A is 0.253 and in population A+B is 0.247 because the change of the allelic frequencies caused by the mixture of the populations tends to break the LD between loci. When more populations are included in the mixture, the degradation of the LD is stronger as in the case of the A+B+C+D population in which the average r is 0.234. The average r declines with the distance between the loci and therefore in population A the average LD of loci within the same chromosome is 0.084 and between chromosomes is 0.045. In population A+B the average LD of loci within the same chromosome is 0.105, 25% more than in population A, and between chromosomes is 0.089, 97% more. The increase of the values of LD of distant loci found in the A+B population occurs because, in the populations A and B, selection and genetic drift forced the frequencies of the loci to extremes. When data from these populations were mixed, the causative mutations and the markers that were fixed oppositely in the two populations appeared with intermediate frequencies that created a high LD over large distances. The A+B+C+D population presents an average LD within-chromosome just 6% (0.089) higher than in population A and between-chromosomes 58% (0.071) higher. The changes of the allelic frequencies in this population also cause the creation of LD between distant loci, but with lower intensity because when mixing 4 populations, it is less probable to encounter loci with extreme opposite frequencies.

At the moment of estimating the effects of the markers, those markers that are in LD with a causative mutation can capture a part of its effect depending on the value of the LD between them. In the case of the A+B population, the fact that many markers are in strong LD with many distant causative mutations, results in these markers capturing effects from those distant loci and therefore, their effects are clearly overestimated, as it can be seen in the figure 8. The breeding values produced with these estimates are less accurate than the ones that are produced by the estimates from the A+B+C+D population because in this population fewer markers present high LD with distant loci.

Figure 9 shows the example of the marker 11437 which is located on the 8th chromosome. This marker was estimated to have a very small effect in the populations A and A+B+C+D but one of the largest in the population A+B. When considering the LD that this marker presents with all the causative mutations, it can be seen that it is very high even with distant loci in the A+B population. Instead, the same marker in the other populations has lower values of LD and with loci in shorter distances.

Table 9. The average LD (r) between markers and causative mutations when they are consecutive, located in the same (within) and different (between) chromosome, in the populations A, A+B and A+B+C+D.

	A	**A+B**	**A+B+C+D**
Average r of consecutive loci	0.253	0.247	0.234
Average r within chromosome	0.084	0.105	0.089
Average r between chromosomes	0.045	0.089	0.071

Figure 8. Graphical representation of the estimates of the marker effects for trait 1 from a) population A, b) population A+B, c) population A+B+C+D.

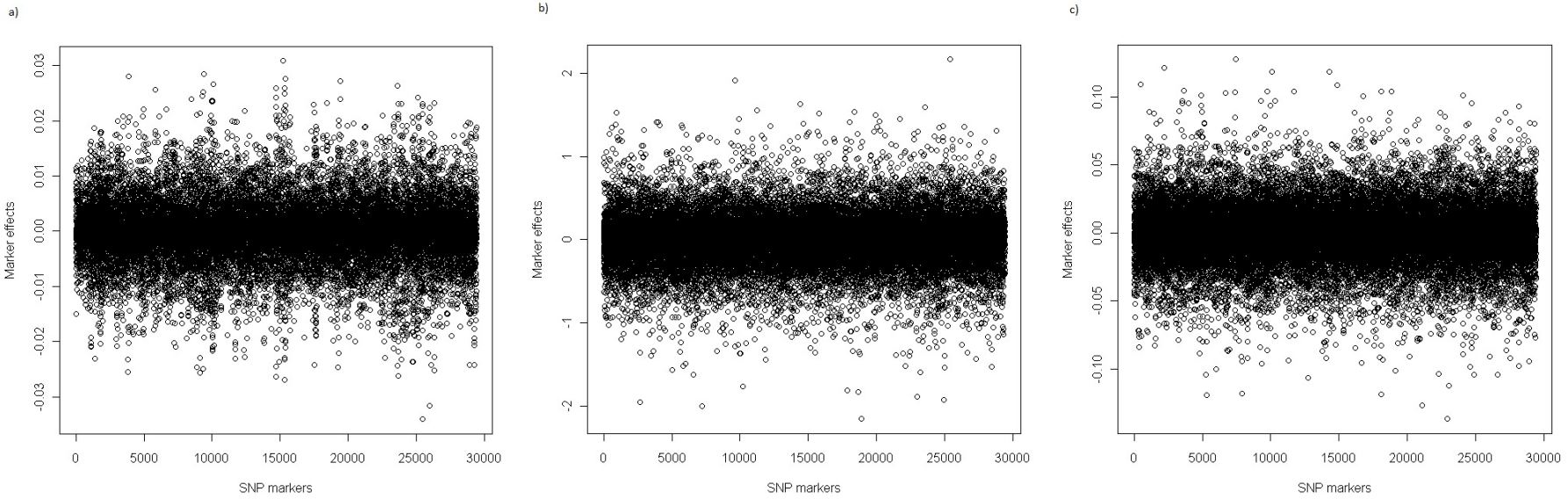
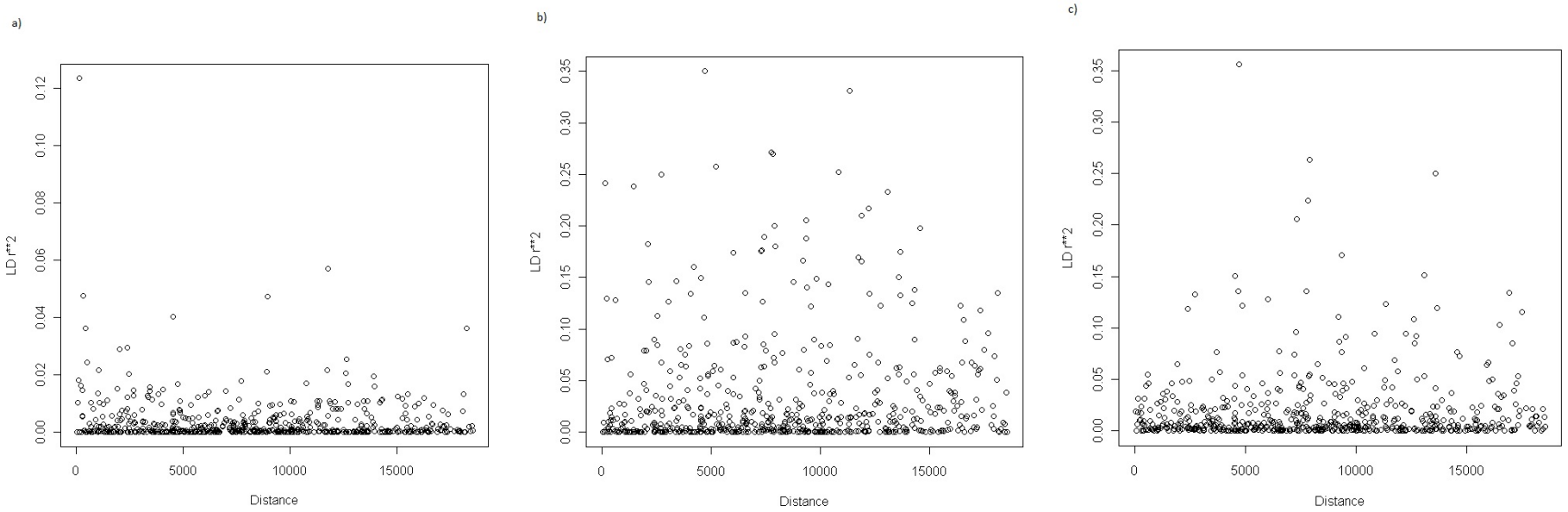


Figure 9. The LD between the marker 11437 and the causative mutations affecting the first trait represented over the distance between them in a) population A, b) population A+B and c) population A+B+C+D.



Marker density

In addition, a second scenario was simulated with a higher density of markers. For this purpose, 30,000 biallelic markers were distributed evenly along only 4 chromosomes of 1 Morgan each, resulting in a density of 75 markers per cM. The scenarios of the number of causative mutations that control the traits and the type and number of generations of divergence were the same as in the first case. The results of the accuracy of the predictions obtained from the purebred population A and the admixed populations A+B and A+B+C+D are presented in Table 10. For simplicity, the results from the other populations are not shown since they are equivalent to those presented.

The accuracies of the purebred populations, when training and validating in the same population, increased around 11% and 7% over the previous case for the traits 1 and 2, respectively. This increase is consistent under all scenarios of number of generations of divergence. The predictive ability of the purebred populations over the others was improved significantly (0.51 and 0.55 for traits 1 and 2, respectively) compared to the case of lower marker density (0.3 and 0.37 for traits 1 and 2, respectively) under the scenario of 5 generations of divergence. Nevertheless, these accuracies declined rapidly to 0.15 (trait 1) and 0.20 (trait 2) and close to 0 with 50 and 200 generations of divergence, respectively.

The higher marker density favored most the predictive ability of the admixed population A+B for the populations that were included in the mixture. The accuracies obtained were increased around 90% and 50% for traits 1 and 2, respectively. The accuracies obtained for the populations that are not included in the admixed population increased as well, being 0.49 for the first trait and 0.56 for the second, with 5 generations of divergence. These accuracies declined with the generations of divergence similarly to the ones obtained from the purebred when predicting another population.

Finally, the accuracies obtained from the A+B+C+D population, comparing with the previous scenario, were increased only 25% for both traits when the populations diverged for only 5 generations but showed a remarkable increase when the populations had diverged for more generations. In the case of 50 generations the increase was of 35% and 70% for traits 1 and 2, respectively and in the case of 200 generations up to 90% and 75%.

The results from the linkage disequilibrium study (Table 11) showed that the persistence of the LD across the populations is almost identical in both cases of marker density, with only a slight increase in the case of the higher marker density (i.e. The admixed population A+B maintains 65.6% and 66.7% of the gene-marker pairs in high LD found in population A in the case of 300 and 100 QTLs, respectively.)

As it can be noticed, the admixed population A+B performed better than the A+B+C+D population in this case, even though the LD patterns are the same as before. This happens due to the fact that the simulated genome consist of only 4 chromosomes and therefore, due to the effect of linkage, the frequencies of the loci in the populations A and B were not allowed to have extreme opposite frequencies by chance.

Overall, the results of the comparison between both scenarios agree with the studies of Toosi et al. (2009) and de Roos et al. (2009), and show that the increased marker density gives higher accuracies when the populations have diverged for only a few generations, benefits more the predictions made in the admixed populations than the ones made in the purebreds and finally, favors the accuracies for traits with low heritability.

Number of QTLs

Further, another scenario with only 100 causative mutations or QTL affecting each trait was also simulated and its predictive ability for all populations was compared with the scenario of 300 genes affecting the traits. The marker density was the same as in the first case. The results (not shown) of the accuracies show that the smaller number of genes affected only the predictive ability of the populations under selection and the admixed populations that contained data from these selected populations. We were not able to found any difference in the accuracy between the 100 and 300 QTL scenarios when the populations are separated for 5 generations, but the accuracy of the selected populations declines faster with the generations of divergence in the 100 genes scenario. Population A showed a decline in the accuracy for the trait 1 of 25% after 200 generations of selection whereas the same reduction in the case of 300 genes was only 15%. Likewise, the accuracy of population B for the trait 2 declined 23% in the case of 100 genes and only 10% in the case of 300 genes. The admixed populations that contain data from the purebred populations under selection showed as well a larger reduction in their accuracies with the generations of evolution in the case of 100 genes. This occurs

because the alleles of a small number of genes are fixed or lost easier during the process of selection and the loss of genetic variance is faster.

However, the study of Kizilkaya et al. (2010) showed that the accuracies of the predicted values depend on the number of genes that control the trait of interest. The accuracies he obtained from training in an admixed population started at 0.388 when there were 50 QTLs and declined to 0.2 when the number of QTLs was 500. It should be noted that this authors compare scenarios with 10 times more QTL whereas in this study the spectrum of scenarios was only a ratio of 3 in the number of genes. Thus, it is probable that the differences between them were too small to be detected with only 10 replicates of simulation.

The findings of this study suggest a promising alternative for the beef cattle industry. There are some important advantages in using an admixed population that consists of data from all available populations. Firstly, the size of the reference population can be easily increased just by adding a few individuals from each of the many existing populations. Secondly, the large number of populations of different genetic origin would lower further the intensity of the phenomenon of overestimation of the marker effects, and finally, it will not be necessary to have different procedures of prediction for every population since the estimated effects from the admixed population are applicable to all populations. This strategy, combined with the use of a high density marker map could result in accurate predictions suitable for the implementation of genomic selection in beef cattle.

Table 10. Accuracy (standard error) of the predicted breeding values of all validation sets obtained from training in the purebred population A and in the admixed populations A+B and A+B+C+D.

		5 Generations			50 Generations			200 Generations		
	Tr* / **Val***	**A**	**A+B**	**A+B+C+D**	**A**	**A+B**	**A+B+C+D**	**A**	**A+B**	**A+B+C+D**
Trait 1 (h^2=0.2)	**A**	0.782 (0.007)	0.671 (0.011)	0.645 (0.015)	0.769 (0.013)	0.611 (0.017)	0.566 (0.018)	0.714 (0.012)	0.510 (0.036)	0.467 (0.025)
	B	0.517 (0.017)	0.688 (0.009)	0.674 (0.015)	0.156 (0.023)	0.660 (0.012)	0.604 (0.018)	0.024 (0.030)	0.619 (0.018)	0.542 (0.024)
	C	0.517 (0.019)	0.489 (0.021)	0.645 (0.023)	0.171 (0.029)	0.163 (0.046)	0.570 (0.025)	-0.014 (0.023)	0.018 (0.028)	0.560 (0.014)
	D	0.498 (0.022)	0.471 (0.017)	0.640 (0.017)	0.140 (0.031)	0.137 (0.025)	0.541 (0.018)	0.004 (0.013)	0.017 (0.022)	0.585 (0.019)
	Tr / **Val**	**A**	**A+B**	**A+B+C+D**	**A**	**A+B**	**A+B+C+D**	**A**	**A+B**	**A+B+C+D**
Trait 2 (h^2=0.4)	**A**	0.871 (0.005)	0.775 (0.008)	0.750 (0.011)	0.871 (0.006)	0.766 (0.008)	0.724 (0.009)	0.875 (0.005)	0.720 (0.026)	0.669 (0.014)
	B	0.554 (0.014)	0.746 (0.008)	0.746 (0.008)	0.142 (0.021)	0.695 (0.018)	0.628 (0.016)	0.025 (0.032)	0.535 (0.026)	0.536 (0.020)
	C	0.556 (0.023)	0.560 (0.024)	0.763 (0.011)	0.266 (0.019)	0.226 (0.031)	0.704 (0.014)	0.023 (0.031)	0.047 (0.018)	0.687 (0.014)
	D	0.587 (0.015)	0.561 (0.013)	0.780 (0.007)	0.251 (0.027)	0.210 (0.032)	0.686 (0.015)	0.042 (0.024)	0.022 (0.030)	0.709 (0.016)

Table 11. Percentage (standard error) of gene-marker pairs in high LD in the validation sets found in the training sets, and correlation (standard error) between the LD of these pairs in the validation sets and the purebred population A and the admixed populations A+B and A+B+C+D.

	5 Generations			50 Generations			200 Generations		
Tr* / Val*	**A**	**A+B**	**A+B+C+D**	**A**	**A+B**	**A+B+C+D**	**A**	**A+B**	**A+B+C+D**
A	100%	66.7% (0.746)	58.8% (1.317)	100%	32.5% (0.469)	17.3% (0.465)	100%	27.2% (0.526)	8.6% (0.248)
	1	0.983 (0.001)	0.973 (0.001)	1	0.904 (0.002)	0.821 (0.003)	1	0.834 (0.003)	0.668 (0.006)
B	49.9% (0.952)	65.3% (0.902)	57.8% (0.920)	11.9% (0.346)	31.6% (0.510)	17.5% (0.628)	2.8% (0.228)	27% (0.514)	8.6% (0.213)
	0.917 (0.004)	0.982 (0.001)	0.972 (0.001)	0.336 (0.009)	0.905 (0.003)	0.820 (0.005)	0.020 (0.007)	0.841 (0.004)	0.694 (0.008)
C	55.8% (1.053)	57.4% (1.039)	64% (0.863)	13.9% (0.444)	16.3% (0.620)	20.7% (0.556)	3.2% (0.119)	10.8% (0.483)	9.5% (0.277)
	0.935 (0.003)	0.952 (0.002)	0.977 (0.001)	0.396 (0.008)	0.485 (0.012)	0.823 (0.003)	0.024 (0.003)	0.054 (0.007)	0.659 (0.007)
D	56.1% (0.738)	57.2% (0.845)	64.1% (0.986)	14.4% (0.628)	16% (0.456)	20.3% (0.585)	3.1% (0.164)	10.8% (0.597)	9.9% (0.323)
	0.936 (0.003)	0.953 (0.003)	0.978 (0.001)	0.400 (0.009)	0.481 (0.007)	0.811 (0.003)	0.015 (0.005)	0.049 (0.006)	0.667 (0.006)

CONCLUSIONS

The conclusions of this study are:

1. Genomic prediction when training and validating in the same population gives high accuracies. However, these accuracies decline with the number of generations when the population is undergoing a process of selection for the trait of interest.

2. The predictive ability of one population over another is low and declines further with the number of generations of divergence.

3. The predicted ability of the admixed populations is better for the populations that are included in the mixture than for the ones that are not included. Nevertheless, these accuracies are remarkably lower than the ones obtained from the purebred populations.

4. When all populations available are represented in the admixed training set the accuracies increase for all populations compared to the accuracies obtained from the admixed x2 populations.

5. A high marker density genotyping benefits the accuracies of all populations but mostly the admixed populations and the predictions for traits with low heritability.

6. The number of QTLs that control the traits affects mainly the predictive ability of the populations that are under selection for these traits. The smaller the number of QTLs the faster the reduction in the accuracy with the number of generations of selection.

7. A genomic prediction procedure from an admixed population generated by several related genetic origins and with a high density map of markers may be useful for selection of all populations involved.

REFERENCES

Cleveland, M., Forni, S., Deeb, N., Maltecca, C. (2010) Genomic breeding value prediction using three Bayesian methods and application to reduced density marker panels. BMC Proceedings 4, S6.

Croiseau, P., Legarra, A., Guillaume, F., Fritz, S., Baur, A., Colombani, C., Robert-Granié, C., Boichard, D., Ducrocq, V. (2011) Fine tuning genomic evaluations in dairy cattle through SNP pre-selection with the Elastic-Net algorithm. Genetics Research 93, 409-417.

de los Campos, G., Naya H., Gianola, D., Crossa, J., Legarra, A., Manfredi, E., Weigel, K., Cotes, J. M. (2009). Predicting quantitative traits with regression models for dense molecular markers and pedigree. Genetics 182:375-385.

de Roos, A.P.W., Hayes, B.J., Goddard, M.E. (2009) Reliability of genomic predictions across multiple populations. Genetics 183, 1545-1553.

Dekkers, J.C.M. (2004) Commercial application of marker- and gene-assisted selection in livestock: Strategies and lessons. Journal of Animal Science 84, E313-E328.

Falconer, D.S. and Mackay, T.F.C. (1996) Introduction to Quantitative Genetics, 4th edn. Longman Group UK, Harlow, England.

Gianola, D., Fernando, R.L., Stella, A. (2006) Genomic-assisted prediction of genetic value with semiparametric procedures. Genetics 173, 1761-1776.

Gianola, D., Okut, H., Weigel, K., Rosa, G. (2011) Predicting complex quantitative traits with Bayesian neural networks: a case study with Jersey caows and wheat. BMC Genetics 12, 87.

González-Recio, O., Gianola, D., Long, N., Weigel, K.A., Rosa, G.J.M., Avendaño, S. (2008) Non-parametric methods for incorporating genomic information into genetic evaluations: An application to mortality in broilers. Genetics 178, 2305-2313.

González-Recio, O., Gianola, D., Rosa, G.J.M., Weigel, K.A., Kranis, A. (2009) Genome-assisted prediction of a quantitative trait measured in parents and progeny: application to food conversion rate in chickens. Genetics Selection Evolution 41, 3.

Habier, D., Fernando, R., Kizilkaya, K., Garrick, D. (2011) Extension of the bayesian alphabet for genomic selection. BMC Bioinformatics 12, 186.

Harris, B.L., Johnson, D.L., Spelman, R.J. (2008) Genomic selection in New Zealand and the implications for national evaluation. Proc. ICAR 36th Session, pp.325-330.

Hayes, B.J., Bowman, P.J., Chamberlain, A.C., Goddard, M.E. (2009) Invited review: Genomic selection in dairy cattle: Progress and challenges. Journal of Dairy Science 92, 433-443.

Hayes, B.J., Bowman, P.J., Chamberlain, A.C., Verbyla, K., Goddard, M.E. (2009) Accuracy of genomic breeding values in multi-breed dairy cattle populations. Genetics Selection Evolution 41, 51.

Henderson, C.R. (1949) Estimation of changes in herd environment [abstract]. Journal of Dairy Science 32, 709.

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

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

Henderson, C.R. (1985) Best linear unbiased prediction of non-additive genetic merits in non-inbred populations. Journal of Animal Science 60, 111-117.

Henderson, C.R. and Quaas, R.L. (1976) Multiple trait evaluation using relatives' records. Journal of Animal Science 43, 1188-1197.

Henderson, C.R. (1975) Best linear unbiased estimation and prediction under a selection model. Biometrics 31, 423-447.

Hoeschele, I. and Van Raden, P.M. (1991) Rapid inverse of dominance relationship matrices for non-inbred populations by including sire by dam subclass effects. Journal of Dairy Science 74, 557-569.

Kizilkaya, K., Fernando, R.L., Garrick, D.J. (2010) Genomic prediction of simulated multibreed and purebred performance using observed fifty thousand single nucleotide polymorphism genotypes. Journal of Animal Science 88, 544-551.

Legarra, A., Ricard, A., Filangi, O. (2012) GS3: Genomic Selection, Gibbs Sampling, Gauss Seidel. http://snp.toulouse.inra.fr/~alegarra/

Legarra, A., Robert-Granié, C., Croiseau, P., Guillaume, F., Fritz, S. (2010) Improved Lasso for genomic selection. Genetics Research 93, 77-87.

Lynch, M. and Walsh, B. (1998) Genetics and analysis of quantitative traits. Sinauer Associates, Inc, Sunderland, U.S.A.

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

Misztal, I., Legarra, A., Aquilar, I. (2009) Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information. Journal of Dairy Science 92, 4648-4655.

Park, T., Casella, G. (2008) The Bayesian Lasso. Journal of the American Statistical Association 103, 681-686.

Quaas, R.L. (1976) Computing the diagonal elements of a large numerator relationship matrix. Biometrics 32, 949-953.

Quaas, R.L. and Pollak, E.J. (1980) Mixed model methodology for farm and ranch beef cattle testing programs. Journal of Animal Science 51, 1277-1287.

Quass, R.L. and Pollak E.J. (1981) Modified equations for sire models with groups. Journal of Dairy Science 64, 1868-1872.

Sun, Y.V. (2010) Multigenic modeling of complex disease by random forests. Adv. Genet. 72, 73-99.

Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58, 267-288.

Toosi, A., Fernando, R.L., Dekkers, J.C.M. (2010) Genomic selection in admixed and croosbred populations. Journal of Animal Science 88, 32-46.

Usai, M.G., Goddard, M.E., Hayes, B.J. (2009) LASSO with cross-validation for genomic selection. Genetics Research 91, 427-436.

Weller, J. I. (1994) Economic Aspects of Animal Breeding. Chapman & Hall, London.

Westell, R.A., Quaas, R.L. and Van Vleck, L.D. (1988) Genetic groups in an animal model. Journal of Dairy Science. 71, 1310-1318.

APPENDIX: SIMULATION PROGRAM

```
program simulation


implicit none
integer *1
mar(100,20,1500,2), marn(100,20,1500,2), marnA(100,20,1500,2), marnB(100,20,1500,2), marnC(100,20,1500,2), marnD(100,20,1500,2), grupoA(100,20,1500,2), grupoB(100,20,1500,2),
grupoC(100,20,1500,2), grupoD(100,20,1500,2), finalA(2000,20,1500,2), finalB(2000,20,1500,2), finalC(2000,20,1500,2), finalD(2000,20,1500,2), finalAB(2000,20,1500,2), finalAC(2000,20,1500,2),
finalAD(2000,20,1500,2), finalBC(2000,20,1500,2), finalBD(2000,20,1500,2), finalCD(2000,20,1500,2), finalABCD(2000,20,1500,2)
integer
 irec(50), irecord(50), iefec1(300,2), iefec2(300,2), ef(600), n(30000),  i, j, k, l, ijk, iale, ip, iv, im, ic, ik, ij, pois, neu, g, marca(30000), nanim, ncro, nmar, nale, nmac, nhem, ngen, nmut, nrec, xmin, imin, imut,
nanim2, ngenes, ngen2, ait
real *8
x1, u, lam, yfenA(100), yfenB(100), var1(300), var2(300), media1, media2, p1base(300,300), p2base(300,300), p1(300,300), p2(300,300), Vr1, Vr2, Va1, Va2, des1, des2, varA1(300), varA2(300), varAf1(300),
varAf2(300), var1B(300), var2B(300), var1Df(300), p1sel(300,300), p2sel(300,300), efec1(300), efec2(300), var1Bf(300), var2Bf(300), var1Cf(300), var2Cf(300), var2Df(300), var1ABf(300), var2ABf(300),
var1ACf(300), var2ACf(300), var1ADf(300), var2ADf(300), var1BCf(300), var2BCf(300), var1BDf(300), var2BDf(300), var1CDf(300), var2CDf(300), var1ABCDf(300), var2ABCDf(300), p1A(300,300),
p2A(300,300), p1B(300,300), p2B(300,300), p1C(300,300), p2C(300,300), p1D(300,300), p2D(300,300), p1AB(300,300), p2AB(300,300), p1AC(300,300), p2AC(300,300), p1AD(300,300), p2AD(300,300),
p1BC(300,300), p2BC(300,300), p1BD(300,300), p2BD(300,300), p1CD(300,300), p2CD(300,300), p1ABCD(300,300), p2ABCD(300,300), pA, pB, pC, pD, pAB, pAC, pAD, pBC, pBD, pCD, pABCD,
bv1A(2000), bv2A(2000), bv1B(2000), bv2B(2000), bv1C(2000), bv2C(2000), bv1D(2000), bv2D(2000), yfen1A(2000), yfen2A(2000), yfen1B(2000), yfen2B(2000), yfen1C(2000), yfen2C(2000),
yfen1D(2000), yfen2D(2000), yfen1AB(2000), yfen2AB(2000), yfen1AC(2000), yfen2AC(2000), yfen1AD(2000), yfen2AD(2000), yfen1BC(2000), yfen2BC(2000), yfen1BD(2000), yfen2BD(2000),
yfen1CD(2000), yfen2CD(2000), yfen1ABCD(2000), yfen2ABCD(2000)
x1=110297.
nanim=100
ncro=20
nmar=1500
nale=2
nmac=50
nhem=50
ngen=1000
nanim2=2000
ngenes=300
ngen2=200

! simulation of the base population
do i=1,nanim
   do j=1,ncro
      do k=1,nmar
         do l=1,nale
                  call unif(x1,u)
                  if (u.lt.0.5) then
           mar(i,j,k,l)=1
                  else
             mar(i,j,k,l)=2
                  end if
                  enddo
      enddo
   enddo
enddo
! simulation of  1000 generations of random mating
do ijk=1,ngen
   do i=1,nanim
      call unif(x1,u)
      ip=int(u*nmac)+1
      call unif(x1,u)
      im=int(u*nhem)+nmac+1
!  father's chromosome
      do j=1,ncro
      call unif(x1,u)
      if (u.lt.0.5) then
                  iale=1
      else
                  iale=2
      endif
      marn(i,j,1,1)=mar(ip,j,1,iale)
      lam=1.
      nrec=pois(lam,x1)        ! selection of the number of recombinations
        do ik=1,nrec
        call unif(x1,u)         ! selection of the location of the recombinations
irec(ik)=int(u*(nmar-1))+1
enddo
        irecord=0
                   do ik=1,nrec
                   xmin=999999
                     do ij=1,nrec
                      if (irec(ij).lt.xmin) then
                   xmin=irec(ij)
                        imin=ij
                     endif
enddo
                     irecord(ik)=xmin
                     irec(imin)=999999
                   enddo
                   irecord(nrec+1)=nmar
                   if (nrec.eq.0) then
                     do k=2,nmar
                     marn(i,j,k,1)=mar(ip,j,k,iale)
                     enddo
                   else
                     ic=2
                     do ik=1,nrec+1
                        do k=ic,irecord(ik)
                        marn(i,j,k,1)=mar(ip,j,k,iale)
                        enddo
                        ic=irecord(ik)+1
                        if (iale.eq.1) then
                                iale=2
                        else
                                iale=1
                        endif
                     enddo
                   endif
!    mutation
                   lam=1.
                   nmut=pois(lam,x1)                        ! selection of the number of mutations
                   do ik=1,nmut
                   call unif(x1,u)
                   imut=int(u*nmar)+1                       ! selection of the location  of the mutation
                   if (marn(i,j,imut,1).eq.1) then
                        marn(i,j,imut,1)=2
                   else
                        marn(i,j,imut,1)=1
                   endif
                   enddo
        enddo
!   mother's chromosome
                   do j=1,ncro
                     call unif(x1,u)
                     if (u.lt.0.5) then
```

```
                    iale=1
                  else
                    iale=2
                  endif
                  marn(i,j,1,2)=mar(im,j,1,iale)
                  lam=1.
                  nrec=pois(lam,x1)          ! selection of the number of recombinations
                  do ik=1,nrec
                     call unif(x1,u)                    ! selection of the location of the recombinations
                irec(ik)=int(u*(nmar-1))+1
                enddo
                  irecord=0
                  do ik=1,nrec
                  xmin=999999
                     do ij=1,nrec
                      if (irec(ij).lt.xmin) then
                                xmin=irec(ij)
                                  imin=ij
                       endif
                enddo
                     irecord(ik)=xmin
                     irec(imin)=999999
                  enddo
                  irecord(nrec+1)=nmar
                  if (nrec.eq.0) then
                    do k=2,nmar
                       marn(i,j,k,2)=mar(im,j,k,iale)
                    enddo
                  else
                    ic=2
                    do ik=1,nrec+1
                      do k=ic,irecord(ik)
                      marn(i,j,k,2)=mar(im,j,k,iale)
                      enddo
                      ic=irecord(ik)+1
                      if (iale.eq.1) then
                                iale=2
                      else
                                iale=1
                      endif
                    enddo
          endif
!    mutation
                  lam=1.
                  nmut=pois(lam,x1)                  ! selection of the number of mutations
                  do ik=1,nmut
                   call unif(x1,u)            ! selection of the location of the mutations
                   imut=int(u*nmar)+1
                   if (marn(i,j,imut,2).eq.1) then
                      marn(i,j,imut,2)=2
                   else
                      marn(i,j,imut,2)=1
                endif
                   enddo
       enddo
    enddo
mar=marn
print *, ijk
enddo
open(11,file='info.txt')

!    Trait 1, h2=0.2, mean=100
! simulation of the causative mutations and their effect for trait 1
open(16,file='effects1.txt')
do i=1,ngenes
   call unif(x1,u)                          ! selection of chromosome
   iefec1(i,1)=int(u*ncro)+1
   call unif(x1,u)                          ! selection of marker
   iefec1(i,2)=int(u*nmar)+1
   call normal(x1,u)                        ! assignment of the effect
   efec1(i)=u*1
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),efec1(i)
end do
close(16)
! frequencies of the causative mutations of trait 1
open(16,file='freq1-base.txt')
do i=1,ngenes
   p1base(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim
     p1base(iefec1(i,1),iefec1(i,2))=p1base(iefec1(i,1),iefec1(i,2))+real(marn(k,iefec1(i,1),iefec1(i,2),1)+marn(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
   p1base(iefec1(i,1),iefec1(i,2))=p1base(iefec1(i,1),iefec1(i,2))/(2*real(nanim))
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1base(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
do i=1,ngenes
    var1(i)=2*p1base(iefec1(i,1),iefec1(i,2))*(1-p1base(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
    Va1=Va1+var1(i)
end do
print *, 'Va1  base =',Va1
write(11,*) 'Va1 base=',Va1
! resisual variance of trait 1
Vr1=(Va1/0.2)-Va1
print *, 'Vr1 =',Vr1
write(11,*) 'Vr1=',Vr1

!    Trait 2, h2=0.4, mean=1000
! simulation of the causative mutations and their effect for trait 2
open(16,file='effects2.txt')
do i=1,ngenes
  call unif(x1,u)                   ! selection of chromosome
  iefec2(i,1)=int(u*ncro)+1
  call unif(x1,u)                   ! selection of marker
  iefec2(i,2)=int(u*nmar)+1
  call normal(x1,u)                         ! assignment of the effect
  efec2(i)=u*1
  write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),efec2(i)
end do
close(16)
! frequencies of the causative mutations of trait 2
open(16,file='freq2-base.txt')
do i=1,ngenes
    p2base(iefec2(i,1),iefec2(i,2))=0
    do k=1,nanim
      p2base(iefec2(i,1),iefec2(i,2))=p2base(iefec2(i,1),iefec2(i,2))+real(marn(k,iefec2(i,1),iefec2(i,2),1)+marn(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2base(iefec2(i,1),iefec2(i,2))=p2base(iefec2(i,1),iefec2(i,2))/(2*real(nanim))
    write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2base(iefec2(i,1),iefec2(i,2))
end do
```

```
close(16)
! additive variance of trait 2
Va2=0
do i=1,ngenes
   var2(i)=2*p2base(iefec2(i,1),iefec2(i,2))*(1-p2base(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2(i)
end do
print *, 'Va2  base =',Va2
write(11,*) 'Va2 base=',Va2
! resisual variance of trait 2
Vr2=(Va2/0.4)-Va2
print *, 'Vr2 =',Vr2
write(11,*) 'Vr2=',Vr2

! write position of causative mutations and neutral markers
do i=1,ngenes
   ef(i)=(iefec1(i,1)-1)*nmar+iefec1(i,2)
   ef(ngenes+i)=(iefec2(i,1)-1)*nmar+iefec2(i,2)
end do
open(16,file='genes.txt')
do i=1,2*ngenes-1
   do j=i+1,2*ngenes
     if (ef(j).lt.ef(i)) then
       k=ef(j)
       ef(j)=ef(i)
       ef(i)=k
     end if
   end do
end do
   ait=0
   do i=1,2*ngenes
      if (ef(i)==ef(i-1)) then
        goto 40
      else
        ait=ait+1
        write(16,*) ait,ef(i)
      end if
40 end do
close(16)
print *, 'N° of genes=',ait
write(11,*) 'N° of genes=',ait
open(16,file='neutral markers.txt')
 neu=0
 do i=1,ncro*nmar
    do j=1,2*ngenes
       if (i==ef(j)) then
                  goto 9
       end if
    end do
    neu=neu+1
    n(neu)=i
    write(16,*) neu,n(neu)
9 end do
 print *, 'N° of neutral markers=',neu
close(16)
write(11,*) 'N° of neutral markers=',neu

!  Population A under selection for trait 1
marnA=marn
! simulation of phenotypes (mean=100)
do i=1,nanim
   yfenA(i)=100
   do j=1,ngenes
   yfenA(i)=yfenA(i)+(marnA(i,iefec1(j,1),iefec1(j,2),1)+marnA(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
   end do
   call normal(x1,u)
   yfenA(i)=yfenA(i)+u*sqrt(Vr1)
end do
! phenotypic mean
media1=0
do i=1,nanim
media1=media1+yfenA(i)
end do
media1=media1/nanim
! phenotypic standard deviation
des1=0
do i=1,nanim
des1=des1+(yfenA(i)-media1)**2
end do
des1=des1/(nanim-1)
des1=sqrt(des1)
print *, 'mean & s.d. trait 1 =',media1,des1
write(11,*) 'mean & s.d =',media1,des1
open(20,file='pheno_ mean_A.txt')
open(21,file='variance_1A.txt')
open(22,file='variance_2A.txt')
! selection for n generations
do ijk=1,ngen2
   do i=1,nanim
   1 call unif(x1,u)
                  ip=int(u*nmac)+1
    if (yfenA(ip).lt.media1-des1) then
                  goto 1
     end if
   2 call unif(x1,u)
                  im=int(u*nhem)+nmac+1
    if (yfenA(im).lt.media1-des1) then
                  goto 2
    end if
! father's chromosome
    do j=1,ncro
       call unif(x1,u)
       if (u.lt.0.5) then
         iale=1
       else
         iale=2
       endif
       grupoA(i,j,1,1)=marnA(ip,j,1,iale)
       lam=1.
       nrec=pois(lam,x1)
       do ik=1,nrec
                  call unif(x1,u)
                  irec(ik)=int(u*(nmar-1))+1
enddo
       irecord=0
          do ik=1,nrec
                  xmin=999999
                     do ij=1,nrec
                        if (irec(ij).lt.xmin) then
                  xmin=irec(ij)
                                imin=ij
```

```
                    endif
                enddo
                    irecord(ik)=xmin
                    irec(imin)=999999
                 enddo
                 irecord(nrec+1)=nmar
        if (nrec.eq.0) then
                    do k=2,nmar
                       grupoA(i,j,k,1)=marnA(ip,j,k,iale)
                    enddo
                 else
                    ic=2
                    do ik=1,nrec+1
                       do k=ic,irecord(ik)
                                grupoA(i,j,k,1)=marnA(ip,j,k,iale)
                enddo
                        ic=irecord(ik)+1
                        if (iale.eq.1) then
                                iale=2
                        else
                                iale=1
                        endif
                      enddo
                    endif
! mutation
                    lam=1.
                    nmut=pois(lam,x1)
                    do ik=1,nmut
                       call unif(x1,u)
                       imut=int(u*nmar)+1
                       if (grupoA(i,j,imut,1).eq.1) then
                          grupoA(i,j,imut,1)=2
                       else
                                grupoA(i,j,imut,1)=1
                       endif
                    enddo
                 enddo
! mother's chromosome
                 do j=1,ncro
                    call unif(x1,u)
                     if (u.lt.0.5) then
                       iale=1
                    else
                       iale=2
                    endif
                    grupoA(i,j,1,2)=marnA(im,j,1,iale)
                    lam=1.
                    nrec=pois(lam,x1)
                    do ik=1,nrec
                       call unif(x1,u)
                       irec(ik)=int(u*(nmar-1))+1
                    enddo
                    irecord=0
                    do ik=1,nrec
                    xmin=999999
                       do ij=1,nrec
                          if (irec(ij).lt.xmin) then
                xmin=irec(ij)
                             imin=ij
                          endif
                enddo
                          irecord(ik)=xmin
                          irec(imin)=999999
                       enddo
                       irecord(nrec+1)=nmar
                       if (nrec.eq.0) then
                          do k=2,nmar
                             grupoA(i,j,k,2)=marnA(im,j,k,iale)
                          enddo
                       else
                          ic=2
                          do ik=1,nrec+1
                             do k=ic,irecord(ik)
                grupoA(i,j,k,2)=marnA(im,j,k,iale)
                                   enddo
                                      ic=irecord(ik)+1
                                      if (iale.eq.1) then
                                         iale=2
                                else
                                         iale=1
                                endif
                          enddo
        endif
! mutation
                    lam=1.
                    nmut=pois(lam,x1)
                    do ik=1,nmut
                       call unif(x1,u)
                       imut=int(u*nmar)+1
                        if (grupoA(i,j,imut,2).eq.1) then
                                   grupoA(i,j,imut,2)=2
                        else
                             grupoA(i,j,imut,2)=1
                       endif
                     enddo
    enddo
  enddo
! simulation of phenotypes
  do i=1,nanim
  yfenA(i)=100
     do j=1,ngenes
        yfenA(i)=yfenA(i)+(grupoA(i,iefec1(j,1),iefec1(j,2),1)+grupoA(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
      end do
      call normal(x1,u)
       yfenA(i)=yfenA(i)+u*sqrt(Vr1)
  end do
! mean
media1=0
do i=1,nanim
    media1=media1+yfenA(i)
end do
media1=media1/nanim
write(20,*) ijk,media1
! standard deviation
des1=0
do i=1,nanim
des1=des1+(yfenA(i)-media1)**2
end do
des1=des1/(nanim-1)
des1=sqrt(des1)
  print *, ijk,media1,des1
```

```
! frequencies of the causative mutations of trait 1
 do i=1,ngenes
 p1(iefec1(i,1),iefec1(i,2))=0
    do k=1,nanim
      p1(iefec1(i,1),iefec1(i,2))=p1(iefec1(i,1),iefec1(i,2))+real(grupoA(k,iefec1(i,1),iefec1(i,2),1)+grupoA(k,iefec1(i,1),iefec1(i,2),2)-2)
    end do
    p1(iefec1(i,1),iefec1(i,2))=p1(iefec1(i,1),iefec1(i,2))/(2*real(nanim))
 end do
! additive variance trait 1
 Va1=0
 varA1(i)=0
 do i=1,ngenes
    varA1(i)=2*p1(iefec1(i,1),iefec1(i,2))*(1-p1(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
    Va1=Va1+varA1(i)
 end do
 !print *, 'additive variance trait 1 popul A=',ijk,Va1
 write(21,*) ijk,Va1,Vr1
! frequencies for causative mutations trait 2
 do i=1,ngenes
    p2(iefec2(i,1),iefec2(i,2))=0
    do k=1,nanim
    p2(iefec2(i,1),iefec2(i,2))=p2(iefec2(i,1),iefec2(i,2))+real(grupoA(k,iefec2(i,1),iefec2(i,2),1)+grupoA(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2(iefec2(i,1),iefec2(i,2))=p2(iefec2(i,1),iefec2(i,2))/(2*real(nanim))
 end do
! additive variance trait 2
Va2=0
varA2(i)=0
do i=1,ngenes
    varA2(i)=2*p2(iefec2(i,1),iefec2(i,2))*(1-p2(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
    Va2=Va2+varA2(i)
end do
write(22,*) ijk,Va2,Vr2
marnA=grupoA
end do
close(20)
close(21)
close(22)

! Final population A
do i=1,nanim2
 3 call unif(x1,u)
   ip=int(u*nmac)+1
   if (yfenA(ip).lt.media1-des1) then
     goto 3
   end if
 4 call unif(x1,u)
   im=int(u*nhem)+nmac+1
   if (yfenA(im).lt.media1-des1) then
     goto 4
   end if
! father's chromosome
  do j=1,ncro
    call unif(x1,u)
    if (u.lt.0.5) then
      iale=1
    else
      iale=2
    endif
    finalA(i,j,1,1)=grupoA(ip,j,1,iale)
    lam=1.
    nrec=pois(lam,x1)
    do ik=1,nrec
call unif(x1,u)
      irec(ik)=int(u*(nmar-1))+1
    enddo
    irecord=0       !desde aqui ordeno
do ik=1,nrec
    xmin=999999
      do ij=1,nrec
       if (irec(ij).lt.xmin) then
                    xmin=irec(ij)
                        imin=ij
       endif
enddo
      irecord(ik)=xmin
      irec(imin)=999999
   enddo
   irecord(nrec+1)=nmar
   if (nrec.eq.0) then
     do k=2,nmar
       finalA(i,j,k,1)=grupoA(ip,j,k,iale)
     enddo
   else
     ic=2
     do ik=1,nrec+1
       do k=ic,irecord(ik)
                    finalA(i,j,k,1)=grupoA(ip,j,k,iale)
enddo
         ic=irecord(ik)+1
         if (iale.eq.1) then
                        iale=2
         else
                        iale=1
         endif
       enddo
  endif
! mutation
  lam=1.
  nmut=pois(lam,x1)
  do ik=1,nmut
      call unif(x1,u)
      imut=int(u*nmar)+1
      if (finalA(i,j,imut,1).eq.1) then
                      finalA(i,j,imut,1)=2
      else
                      finalA(i,j,imut,1)=1
      endif
   enddo
 enddo
!   mother's chromosome
  do j=1,ncro
    call unif(x1,u)
    if (u.lt.0.5) then
      iale=1
    else
      iale=2
    endif
    finalA(i,j,1,2)=grupoA(im,j,1,iale)
    lam=1.
    nrec=pois(lam,x1)
```

```
    do ik=1,nrec
     call unif(x1,u)
     irec(ik)=int(u*(nmar-1))+1
    enddo
    irecord=0
     do ik=1,nrec
     xmin=999999
       do ij=1,nrec
        if (irec(ij).lt.xmin) then
xmin=irec(ij)
                        imin=ij
         endif
enddo
        irecord(ik)=xmin
        irec(imin)=999999
      enddo
      irecord(nrec+1)=nmar
      if (nrec.eq.0) then
        do k=2,nmar
          finalA(i,j,k,2)=grupoA(im,j,k,iale)
        enddo
      else
        ic=2
        do ik=1,nrec+1
          do k=ic,irecord(ik)
                    finalA(i,j,k,2)=grupoA(im,j,k,iale)
                    enddo
                     ic=irecord(ik)+1
                     if (iale.eq.1) then
                        iale=2
                     else
                        iale=1
                     endif
        enddo
    endif
! mutation
        lam=1.
        nmut=pois(lam,x1)
        do ik=1,nmut
         call unif(x1,u)
                    imut=int(u*nmar)+1
          if (finalA(i,j,imut,2).eq.1) then
                     finalA(i,j,imut,2)=2
                    else
             finalA(i,j,imut,2)=1
                    endif
        enddo
   enddo
enddo
! frequencies of the causative mutations of trait 1
open(16,file='freq1-A.txt')
do i=1,ngenes
 p1A(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
     p1A(iefec1(i,1),iefec1(i,2))=p1A(iefec1(i,1),iefec1(i,2))+real(finalA(k,iefec1(i,1),iefec1(i,2),1)+finalA(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
   p1A(iefec1(i,1),iefec1(i,2))=p1A(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1A(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
varAf1=0
do i=1,ngenes
    varAf1(i)=2*p1A(iefec1(i,1),iefec1(i,2))*(1-p1A(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
    Va1=Va1+varAf1(i)
end do
print *, 'Variance trait 1 population A=',Va1
write(11,*) 'Variance trait 1 population A=',Va1
! frequencies of the causative mutation of trait 2
open(16,file='freq2-A.txt')
do i=1,ngenes
   p2A(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
      p2A(iefec2(i,1),iefec2(i,2))=p2A(iefec2(i,1),iefec2(i,2))+real(finalA(k,iefec2(i,1),iefec2(i,2),1)+finalA(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2A(iefec2(i,1),iefec2(i,2))=p2A(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2A(iefec2(i,1),iefec2(i,2))
 end do
close(16)
! additive variance trait 2
Va2=0
varAf2=0
do i=1,ngenes
    varAf2(i)=2*p2A(iefec2(i,1),iefec2(i,2))*(1-p2A(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
     Va2=Va2+varAf2(i)
end do
print *, 'Variance trait 2 population A=',Va2
write(11,*) 'Variance trait 2 pooulation A=',Va2
! simulation of phenotypes
open(13,file='f1A.txt')
open(14,file='f2A.txt')
do i=1,nanim2
yfen1A(i)=100
   do j=1,ngenes
      yfen1A(i)=yfen1A(i)+(finalA(i,iefec1(j,1),iefec1(j,2),1)+finalA(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
    end do
    call normal(x1,u)
    yfen1A(i)=yfen1A(i)+u*sqrt(Vr1)
    write(13,*) yfen1A(i),i,1
end do
close(13)
do i=1,nanim2
yfen2A(i)=1000
   do j=1,ngenes
      yfen2A(i)=yfen2A(i)+(finalA(i,iefec2(j,1),iefec2(j,2),1)+finalA(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
   end do
   call normal(x1,u)
   yfen2A(i)=yfen2A(i)+u*sqrt(Vr2)
   write(14,*) yfen2A(i),i,1
end do
close(14)
! write genotypes
open(13,file='gA.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                write(13,*) i,((j-1)*nmar+k),finalA(i,j,k,1),finalA(i,j,k,2)
      end do
   end do
end do
```

```
close(13)
! marker and gene frequencies
open(13,file='pA.txt')
do i=1,ncro
   do j=1,nmar
   pA=0
     do k=1,nanim2
       pA=pA+real(finalA(k,i,j,1)+finalA(k,i,j,2)-2)
     end do
     write(13,*) (i-1)*nmar+j,pA/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nA.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                  do g=1,ngenes
                  if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 25
                  end if
                  if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 25
                  end if
                  end do
                  write(13,*) i,(j-1)*nmar+k,(finalA(i,j,k,1)+finalA(i,j,k,2)-3)
25    end do
    end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nA.txt')
open(13,file='genotA.txt')
do i=1,nanim2
marca(j)=0
   do j=1,neu
   read(20,*) ip,im,iv
   marca(j)=iv+1
   end do
   write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)
! true breeding values for traits 1 & 2
open(14,file='tbv1A.txt')
open(15,file='tbv2A.txt')
do i=1,nanim2
bv1A(i)=0
   do j=1,ngenes
     bv1A(i)=bv1A(i)+(finalA(i,iefec1(j,1),iefec1(j,2),1)+finalA(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
   end do
  write(14,*) i,bv1A(i)
end do
close(14)
do i=1,nanim2
bv2A(i)=0
   do j=1,ngenes
     bv2A(i)=bv2A(i)+(finalA(i,iefec2(j,1),iefec2(j,2),1)+finalA(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
   end do
   write(15,*) i,bv2A(i)
end do
close(15)

! Population B under selection for trait 2
marnB=marn
! simulation of phenotypes (media=1000)
do i=1,nanim
yfenB(i)=1000
   do j=1,ngenes
     yfenB(i)=yfenB(i)+(marnB(i,iefec2(j,1),iefec2(j,2),1)+marnB(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
   end do
   call normal(x1,u)
   yfenB(i)=yfenB(i)+u*sqrt(Vr2)
end do
! phenotypic mean
media2=0
do i=1,nanim
media2=media2+yfenB(i)
end do
media2=media2/nanim
! phenotypic standard deviation
des2=0
do i=1,nanim
des2=des2+(yfenB(i)-media2)**2
end do
des2=des2/(nanim-1)
des2=sqrt(des2)
print *, 'mean & s.d.  trait 2 base =',media2,des2
write(11,*) 'mean & s.d. trait 2 base=',media2,des2
open(20,file='pheno_mean_B.txt')
open(21,file='variance_1B.txt')
open(22,file='variance_2B.txt')
! selection for n generations
do ijk=1,ngen2
   do i=1,nanim
   5 call unif(x1,u)
                  ip=int(u*nmac)+1
     if (yfenB(ip).lt.media2-des2) then
                  goto 5
     end if
   6 call unif(x1,u)
                  im=int(u*nhem)+nmac+1
     if (yfenB(im).lt.media2-des2) then
                  goto 6
                end if
! father's chromosome
     do j=1,ncro
     call unif(x1,u)
     if (u.lt.0.5) then
       iale=1
     else
       iale=2
     endif
     grupoB(i,j,1,1)=marnB(ip,j,1,iale)
     lam=1.
     nrec=pois(lam,x1)
     do ik=1,nrec
                  call unif(x1,u)
                  irec(ik)=int(u*(nmar-1))+1
enddo
```

```
        irecord=0
        do ik=1,nrec
        xmin=999999
                        do ij=1,nrec
                        if (irec(ij).lt.xmin) then
                       xmin=irec(ij)
                          imin=ij
                        endif
                       enddo
                        irecord(ik)=xmin
                        irec(imin)=999999
        enddo
         irecord(nrec+1)=nmar
         if (nrec.eq.0) then
           do k=2,nmar
                         grupoB(i,j,k,1)=marnB(ip,j,k,iale)
                       enddo
         else
            ic=2
                        do ik=1,nrec+1
                           do k=ic,irecord(ik)
                             grupoB(i,j,k,1)=marnB(ip,j,k,iale)
                           enddo
                            ic=irecord(ik)+1
                            if (iale.eq.1) then
                              iale=2
                           else
                              iale=1
                           endif
                         enddo
        endif
! mutation
         lam=1.
                       nmut=pois(lam,x1)
                       do ik=1,nmut
                        call unif(x1,u)
                        imut=int(u*nmar)+1
                        if (grupoB(i,j,imut,1).eq.1) then
                           grupoB(i,j,imut,1)=2
                        else
                           grupoB(i,j,imut,1)=1
                        endif
                       enddo
      enddo
! mother's chromosome
      do j=1,ncro
       call unif(x1,u)
       if (u.lt.0.5) then
         iale=1
       else
         iale=2
       endif
       grupoB(i,j,1,2)=marnB(im,j,1,iale)
       lam=1.
       nrec=pois(lam,x1)
       do ik=1,nrec
       call unif(x1,u)
irec(ik)=int(u*(nmar-1))+1
enddo
        irecord=0
       do ik=1,nrec
        xmin=999999
           do ij=1,nrec
                        if (irec(ij).lt.xmin) then
                       xmin=irec(ij)
                            imin=ij
                          endif
                       enddo
                       irecord(ik)=xmin
                       irec(imin)=999999
        enddo
       irecord(nrec+1)=nmar
       if (nrec.eq.0) then
         do k=2,nmar
         grupoB(i,j,k,2)=marnB(im,j,k,iale)
         enddo
       else
          ic=2
          do ik=1,nrec+1
             do k=ic,irecord(ik)
                          grupoB(i,j,k,2)=marnB(im,j,k,iale)
                       enddo
                       ic=irecord(ik)+1
                       if (iale.eq.1) then
                          iale=2
                       else
                          iale=1
                       endif
         enddo
       endif
! mutation
         lam=1.
         nmut=pois(lam,x1)
         do ik=1,nmut
         call unif(x1,u)
         imut=int(u*nmar)+1
         if (grupoB(i,j,imut,2).eq.1) then
                       grupoB(i,j,imut,2)=2
       else
                       grupoB(i,j,imut,2)=1
       endif
       enddo
    enddo
enddo
! simulation of phenotypes
do i=1,nanim
yfenB(i)=1000
   do j=1,ngenes
    yfenB(i)=yfenB(i)+(grupoB(i,iefec2(j,1),iefec2(j,2),1)+grupoB(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
   end do
   call normal(x1,u)
   yfenB(i)=yfenB(i)+u*sqrt(Vr2)
  end do
! phenotypic mean
  media2=0
  do i=1,nanim
  media2=media2+yfenB(i)
  end do
media2=media2/nanim
  write(20,*) ijk,media2
! phenotypic standard deviation
```

```
  des2=0
  do i=1,nanim
des2=des2+(yfenB(i)-media2)**2
end do
  des2=des2/(nanim-1)
des2=sqrt(des2)
  print *, ijk,media2,des2
! frequencies of the causative mutations of trait 1
  do i=1,ngenes
  p1sel(iefec1(i,1),iefec1(i,2))=0
     do k=1,nanim
      p1sel(iefec1(i,1),iefec1(i,2))=p1sel(iefec1(i,1),iefec1(i,2))+real(grupoB(k,iefec1(i,1),iefec1(i,2),1)+grupoB(k,iefec1(i,1),iefec1(i,2),2)-2)
     end do
     p1sel(iefec1(i,1),iefec1(i,2))=p1sel(iefec1(i,1),iefec1(i,2))/(2*real(nanim))
  end do
! additive variance of trait 1
  Va1=0
  var1B(i)=0
  do i=1,ngenes
     var1B(i)=2*p1sel(iefec1(i,1),iefec1(i,2))*(1-p1sel(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
     Va1=Va1+var1B(i)
  end do
  write(21,*) ijk,Va1
! frequencies of the causative mutations of trait 2
  do i=1,ngenes
  p2sel(iefec2(i,1),iefec2(i,2))=0
     do k=1,nanim
      p2sel(iefec2(i,1),iefec2(i,2))=p2sel(iefec2(i,1),iefec2(i,2))+real(grupoB(k,iefec2(i,1),iefec2(i,2),1)+grupoB(k,iefec2(i,1),iefec2(i,2),2)-2)
     end do
     p2sel(iefec2(i,1),iefec2(i,2))=p2sel(iefec2(i,1),iefec2(i,2))/(2*real(nanim))
  end do
! additive variance of trait 2
  Va2=0
  var2B(i)=0
  do i=1,ngenes
     var2B(i)=2*p2sel(iefec2(i,1),iefec2(i,2))*(1-p2sel(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
     Va2=Va2+var2B(i)
  end do
  write(22,*) ijk,Va2
  marnB=grupoB
end do
close(20)
close(21)
close(22)

! Final population B
do i=1,nanim2
 7 call unif(x1,u)
   ip=int(u*nmac)+1
   if (yfenB(ip).lt.media2-des2) then
   goto 7
   end if
 8 call unif(x1,u)
   im=int(u*nhem)+nmac+1
   if (yfenB(im).lt.media2-des2) then
   goto 8
   end if
! father's chromosome
   do j=1,ncro
     call unif(x1,u)
     if (u.lt.0.5) then
       iale=1
     else
       iale=2
     endif
     finalB(i,j,1,1)=grupoB(ip,j,1,iale)
     lam=1.
     nrec=pois(lam,x1)
     do ik=1,nrec
       call unif(x1,u)
irec(ik)=int(u*(nmar-1))+1
enddo
     irecord=0
     do ik=1,nrec
     xmin=999999
       do ij=1,nrec
         if (irec(ij).lt.xmin) then
                        xmin=irec(ij)
                          imin=ij
         endif
enddo
       irecord(ik)=xmin
       irec(imin)=999999
     enddo
     irecord(nrec+1)=nmar
     if (nrec.eq.0) then
       do k=2,nmar
       finalB(i,j,k,1)=grupoB(ip,j,k,iale)
       enddo
     else
       ic=2
       do ik=1,nrec+1
         do k=ic,irecord(ik)
                          finalB(i,j,k,1)=grupoB(ip,j,k,iale)
         enddo
         ic=irecord(ik)+1
         if (iale.eq.1) then
                           iale=2
         else
                           iale=1
         endif
       enddo
   endif
 ! mutar este cromosoma
    lam=1.
    nmut=pois(lam,x1)
do ik=1,nmut
        call unif(x1,u)
         imut=int(u*nmar)+1
         if (finalB(i,j,imut,1).eq.1) then
                          finalB(i,j,imut,1)=2
         else
                          finalB(i,j,imut,1)=1
         endif
     enddo
   enddo
!   mother's chromosome
   do j=1,ncro
     call unif(x1,u)
     if (u.lt.0.5) then
       iale=1
```

```
    else
      iale=2
    endif
    finalB(i,j,1,2)=grupoB(im,j,1,iale)
    lam=1.
    nrec=pois(lam,x1)
    do ik=1,nrec
     call unif(x1,u)
irec(ik)=int(u*(nmar-1))+1
enddo
    irecord=0
     do ik=1,nrec
     xmin=999999
       do ij=1,nrec
        if (irec(ij).lt.xmin) then
xmin=irec(ij)
                    imin=ij
        endif
enddo
       irecord(ik)=xmin
       irec(imin)=999999
     enddo
     irecord(nrec+1)=nmar
     if (nrec.eq.0) then
       do k=2,nmar
        finalB(i,j,k,2)=grupoB(im,j,k,iale)
       enddo
     else
                    ic=2
                    do ik=1,nrec+1
                       do k=ic,irecord(ik)
                         finalB(i,j,k,2)=grupoB(im,j,k,iale)
                       enddo
                       ic=irecord(ik)+1
                       if (iale.eq.1) then
                          iale=2
                       else
                          iale=1
                       endif
                    enddo
     endif
! mutation
     lam=1.
     nmut=pois(lam,x1)
     do ik=1,nmut
       call unif(x1,u)
                    imut=int(u*nmar)+1
       if (finalB(i,j,imut,2).eq.1) then
                      finalB(i,j,imut,2)=2
       else
          finalB(i,j,imut,2)=1
       endif
     enddo
  enddo
enddo
! frequencies of the causative mutations of trait 2
open(16,file='freq2-B.txt')
do i=1,ngenes
   p2B(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
      p2B(iefec2(i,1),iefec2(i,2))=p2B(iefec2(i,1),iefec2(i,2))+ real(finalB(k,iefec2(i,1),iefec2(i,2),1)+finalB(k,iefec2(i,1),iefec2(i,2),2)-2)
   end do
   p2B(iefec2(i,1),iefec2(i,2))=p2B(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2B(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance of trait 2
Va2=0
var2Bf(i)=0
do i=1,ngenes
  var2Bf(i)=2*p2B(iefec2(i,1),iefec2(i,2))*(1-p2B(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
  Va2=Va2+var2Bf(i)
end do
print *, 'Varaince trait 2 population B=',Va2
write(11,*) 'Variance trait 2 population B=',Va2
! frequencies of the causative mutations of trait 1
open(16,file='freq1-B.txt')
do i=1,ngenes
p1B(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
    p1B(iefec1(i,1),iefec1(i,2))=p1B(iefec1(i,1),iefec1(i,2))+real(finalB(k,iefec1(i,1),iefec1(i,2),1)+finalB(k,iefec1(i,1),iefec1(i,2),2)-2)
    end do
    p1B(iefec1(i,1),iefec1(i,2))=p1B(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1B(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1Bf(i)=0
do i=1,ngenes
  var1Bf(i)=2*p1B(iefec1(i,1),iefec1(i,2))*(1-p1B(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
  Va1=Va1+var1Bf(i)
end do
print *, 'Varaince trait 1 population B=',Va1
write(11,*) 'Variance trait 1 population B=',Va1
! simulation of phenotypes
open(13,file='f2B.txt')
open(15,file='f1B.txt')
do i=1,nanim2
yfen2B(i)=1000
   do j=1,ngenes
   yfen2B(i)=yfen2B(i)+(finalB(i,iefec2(j,1),iefec2(j,2),1)+finalB(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
   end do
   call normal(x1,u)
   yfen2B(i)=yfen2B(i)+u*sqrt(Vr2)
   write(13,*) yfen2B(i),i,1
end do
close(13)
do i=1,nanim2
yfen1B(i)=100
   do j=1,ngenes
   yfen1B(i)=yfen1B(i)+(finalB(i,iefec1(j,1),iefec1(j,2),1)+finalB(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
   end do
   call normal(x1,u)
   yfen1B(i)=yfen1B(i)+u*sqrt(Vr1)
   write(15,*) yfen1B(i),i,1
end do
close(15)
! write genotypes
open(13,file='gB.txt')
do i=1,nanim2
```

```
  do j=1,ncro
    do k=1,nmar
                    write(13,*) i,((j-1)*nmar+k),finalB(i,j,k,1),finalB(i,j,k,2)
                    end do
  end do
end do
close(13)
! marker and gene frequencies
open(13,file='pB.txt')
do i=1,ncro
   do j=1,nmar
   pB=0
     do k=1,nanim2
                    pB=pB+real(finalB(k,i,j,1)+finalB(k,i,j,2)-2)
     end do
     write(13,*) (i-1)*nmar+j,pB/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nB.txt')
do i=1,nanim2
  do j=1,ncro
    do k=1,nmar
       do g=1,ngenes
        if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 26
        end if
        if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 26
        end if
       end do
       write(13,*) i,(j-1)*nmar+k,(finalB(i,j,k,1)+finalB(i,j,k,2)-3)
26  end do
  end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nB.txt')
open(13,file='genotB.txt')
do i=1,nanim2
marca(j)=0
   do j=1,neu
    read(20,*) ip,im,iv
    marca(j)=iv+1
   end do
   write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)
! true breeding values for traits 1 & 2
open(14,file='1-tbv2B.txt')
open(15,file='1-tbv1B.txt')
do i=1,nanim2
bv2B(i)=0
   do j=1,ngenes
    bv2B(i)=bv2B(i)+(finalB(i,iefec2(j,1),iefec2(j,2),1)+finalB(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
   end do
   write(14,*) i,bv2B(i)
end do
close(14)
do i=1,nanim2
bv1B(i)=0
   do j=1,ngenes
   bv1B(i)=bv1B(i)+(finalB(i,iefec1(j,1),iefec1(j,2),1)+finalB(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
   end do
   write(15,*) i,bv1B(i)
end do
close(15)

! population C under random mating
marnC=marn
! random mating for n generations
  do ijk=1,ngen2
       do i=1,nanim
                    call unif(x1,u)
                    ip=int(u*nmac)+1
                    call unif(x1,u)
                    im=int(u*nhem)+nmac+1
!   father's chromosome
                    do j=1,ncro
                      call unif(x1,u)
                      if (u.lt.0.5) then
                         iale=1
                       else
                         iale=2
                      endif
                      grupoC(i,j,1,1)=marnC(ip,j,1,iale)
                      lam=1.
                      nrec=pois(lam,x1)
                      do ik=1,nrec
                                    call unif(x1,u)
                                  irec(ik)=int(u*(nmar-1))+1
                    enddo
                      irecord=0
                      do ik=1,nrec
                        xmin=999999
                          do ij=1,nrec
                                      if (irec(ij).lt.xmin) then
                                    xmin=irec(ij)
                                         imin=ij
                                      endif
                    enddo
                          irecord(ik)=xmin
                          irec(imin)=999999
                      enddo
                      irecord(nrec+1)=nmar
                      if (nrec.eq.0) then
                        do k=2,nmar
                        grupoC(i,j,k,1)=marnC(ip,j,k,iale)
                        enddo
              else
                        ic=2
                        do ik=1,nrec+1
                          do k=ic,irecord(ik)
                                      grupoC(i,j,k,1)=marnC(ip,j,k,iale)
                          enddo
                          ic=irecord(ik)+1
                          if (iale.eq.1) then
                             iale=2
                          else
```

```
                        iale=1
                      endif
                      enddo
                    endif
! mutation
                    lam=1.
                    nmut=pois(lam,x1)
                    do ik=1,nmut
                                call unif(x1,u)
                                imut=int(u*nmar)+1
                                if (grupoC(i,j,imut,1).eq.1) then
                                  grupoC(i,j,imut,1)=2
                                 else
                                  grupoC(i,j,imut,1)=1
                                endif
                    enddo
        enddo
! mother's chromosome
                  do j=1,ncro
                    call unif(x1,u)
                    if (u.lt.0.5) then
                      iale=1
                    else
                      iale=2
                    endif
                    grupoC(i,j,1,2)=marnC(im,j,1,iale)
                    lam=1.
                    nrec=pois(lam,x1)
                    do ik=1,nrec
                      call unif(x1,u)
                  irec(ik)=int(u*(nmar-1))+1
                    enddo
                      irecord=0
                    do ik=1,nrec
                    xmin=999999
                      do ij=1,nrec
                      if (irec(ij).lt.xmin) then
                                  xmin=irec(ij)
                                    imin=ij
                        endif
                  enddo
                        irecord(ik)=xmin
                        irec(imin)=999999
                    enddo
                    irecord(nrec+1)=nmar
                    if (nrec.eq.0) then
                      do k=2,nmar
                      grupoC(i,j,k,2)=marnC(im,j,k,iale)
                      enddo
                    else
                      ic=2
                      do ik=1,nrec+1
                        do k=ic,irecord(ik)
                        grupoC(i,j,k,2)=marnC(im,j,k,iale)
                      enddo
                       ic=irecord(ik)+1
                       if (iale.eq.1) then
                                    iale=2
                       else
                                    iale=1
                       endif
                      enddo
        endif
! mutation
                     lam=1.
                     nmut=pois(lam,x1)
                     do ik=1,nmut
                      call unif(x1,u)
                       imut=int(u*nmar)+1
                       if (grupoC(i,j,imut,2).eq.1) then
                          grupoC(i,j,imut,2)=2
                       else
                          grupoC(i,j,imut,2)=1
                     endif
                      enddo
      enddo
    enddo
    marnC=grupoC
print*, ijk
end do

! Final population C
do i=1,nanim2
    call unif(x1,u)
    ip=int(u*nmac)+1
    call unif(x1,u)
     im=int(u*nhem)+nmac+1
!   father's chromosoma
      do j=1,ncro
        call unif(x1,u)
         if (u.lt.0.5) then
                    iale=1
         else
                    iale=2
         endif
         finalC(i,j,1,1)=grupoC(ip,j,1,iale)
         lam=1.
         nrec=pois(lam,x1)
         do ik=1,nrec
         call unif(x1,u)
irec(ik)=int(u*(nmar-1))+1
enddo
          irecord=0
         do ik=1,nrec
         xmin=999999
                    do ij=1,nrec
                      if (irec(ij).lt.xmin) then
                    xmin=irec(ij)
                      imin=ij
                     endif
                    enddo
                    irecord(ik)=xmin
                    irec(imin)=999999
       enddo
       irecord(nrec+1)=nmar
       if (nrec.eq.0) then
          do k=2,nmar
           finalC(i,j,k,1)=grupoC(ip,j,k,iale)
                    enddo
       else
                     ic=2
```

```
          do ik=1,nrec+1
                        do k=ic,irecord(ik)
                         finalC(i,j,k,1)=grupoC(ip,j,k,iale)
                        enddo
                         ic=irecord(ik)+1
                         if (iale.eq.1) then
                            iale=2
                         else
                            iale=1
                         endif
                       enddo
       endif
! mutation
        lam=1.
        nmut=pois(lam,x1)
        do ik=1,nmut
                       call unif(x1,u)
                       imut=int(u*nmar)+1
                       if (finalC(i,j,imut,1).eq.1) then
                          finalC(i,j,imut,1)=2
                       else
                          finalC(i,j,imut,1)=1
                       endif
        enddo
     enddo
! mother's chromosome
     do j=1,ncro
        call unif(x1,u)
        if (u.lt.0.5) then
                         iale=1
        else
                         iale=2
        endif
        finalC(i,j,1,2)=grupoC(im,j,1,iale)
         lam=1.
         nrec=pois(lam,x1)
         do ik=1,nrec
                       call unif(x1,u)
                       irec(ik)=int(u*(nmar-1))+1
enddo
          irecord=0
          do ik=1,nrec
           xmin=999999
             do ij=1,nrec
                         if (irec(ij).lt.xmin) then
                       xmin=irec(ij)
                            imin=ij
                         endif
                       enddo
                         irecord(ik)=xmin
                         irec(imin)=999999
          enddo
          irecord(nrec+1)=nmar
          if (nrec.eq.0) then
                         do k=2,nmar
                         finalC(i,j,k,2)=grupoC(im,j,k,iale)
                         enddo
          else
                         ic=2
                         do ik=1,nrec+1
                            do k=ic,irecord(ik)
                              finalC(i,j,k,2)=grupoC(im,j,k,iale)
                            enddo
                            ic=irecord(ik)+1
                            if (iale.eq.1) then
                               iale=2
                            else
                               iale=1
                            endif
                         enddo
          endif
! mutation
          lam=1.
          nmut=pois(lam,x1)
          do ik=1,nmut
           call unif(x1,u)
           imut=int(u*nmar)+1
           if (finalC(i,j,imut,2).eq.1) then
                         finalC(i,j,imut,2)=2
           else
                finalC(i,j,imut,2)=1
           endif
           enddo
   enddo
enddo
! frequencies of the causative mutations of trait 1
open(16,file='freq1-C.txt')
do i=1,ngenes
   p1C(iefec1(i,1),iefec2(i,2))=0
   do k=1,nanim2
     p1C(iefec1(i,1),iefec1(i,2))=p1C(iefec1(i,1),iefec1(i,2))+real(finalC(k,iefec1(i,1),iefec1(i,2),1)+finalC(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
   p1C(iefec1(i,1),iefec1(i,2))=p1C(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1C(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance trait 1
Va1=0
var1Cf(i)=0
do i=1,ngenes
  var1Cf(i)=2*p1C(iefec1(i,1),iefec1(i,2))*(1-p1C(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
  Va1=Va1+var1Cf(i)
end do
print *, 'Variance trait 1 population C=',Va1
write(11,*) 'Variance trait 1 population C=',Va1
! frequencies of the causative mutations for trait 2
open(16,file='freq2-C.txt')
do i=1,ngenes
    p2C(iefec2(i,1),iefec2(i,2))=0
    do k=1,nanim2
    p2C(iefec2(i,1),iefec2(i,2))=p2C(iefec2(i,1),iefec2(i,2))+real(finalC(k,iefec2(i,1),iefec2(i,2),1)+finalC(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2C(iefec2(i,1),iefec2(i,2))=p2C(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2C(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance trait 2
Va2=0
var2Cf(i)=0
do i=1,ngenes
  var2Cf(i)=2*p2C(iefec2(i,1),iefec2(i,2))*(1-p2C(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
```

```
  Va2=Va2+var2Cf(i)
end do
print *, 'Variance trait 2 population C=',Va2
write(11,*) 'Variance trait 2 population C=',Va2
! simulation of phenotypes
open(15,file='f1C.txt')
open(16,file='f2C.txt')
do i=1,nanim2
yfen1C(i)=100
    do j=1,ngenes
    yfen1C(i)=yfen1C(i)+(finalC(i,iefec1(j,1),iefec1(j,2),1)+finalC(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
    end do
    call normal(x1,u)
     yfen1C(i)=yfen1C(i)+u*sqrt(Vr1)
    write(15,*) yfen1C(i),i,1
end do
close(15)
do i=1,nanim2
yfen2C(i)=1000
    do j=1,ngenes
    yfen2C(i)=yfen2C(i)+(finalC(i,iefec2(j,1),iefec2(j,2),1)+finalC(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
    end do
    call normal(x1,u)
    yfen2C(i)=yfen2C(i)+u*sqrt(Vr2)
    write(16,*) yfen2C(i),i,1
end do
close(16)
! write genotypes
open(14,file='gC.txt')
do i=1,nanim2
    do j=1,ncro
      do k=1,nmar
      write(14,*) i,((j-1)*nmar+k),finalC(i,j,k,1),finalC(i,j,k,2)
      end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pC.txt')
do i=1,ncro
    do j=1,nmar
    pC=0
      do k=1,nanim2
                  pC=pC+real(finalC(k,i,j,1)+finalC(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pC/(2*real(nanim2))
    end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nC.txt')
do i=1,nanim2
    do j=1,ncro
       do k=1,nmar
                  do g=1,ngenes
                  if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 27
                  end if
                  if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 27
                  end if
                  end do
                  write(13,*) i,(j-1)*nmar+k,(finalC(i,j,k,1)+finalC(i,j,k,2)-3)
27    end do
    end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nC.txt')
open(13,file='genotC.txt')
do i=1,nanim2
marca(j)=0
    do j=1,neu
    read(20,*) ip,im,iv
    marca(j)=iv+1
    end do
    write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)
! true breeding values for traits 1 & 2
open(14,file='tbv2C.txt')
open(15,file='tbv1C.txt')
do i=1,nanim2
bv2C(i)=0
    do j=1,ngenes
     bv2C(i)=bv2C(i)+(finalC(i,iefec2(j,1),iefec2(j,2),1)+finalC(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
    end do
    write(14,*) i,bv2C(i)
end do
close(14)
do i=1,nanim2
bv1C(i)=0
    do j=1,ngenes
     bv1C(i)=bv1C(i)+(finalC(i,iefec1(j,1),iefec1(j,2),1)+finalC(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
    end do
    write(15,*) i,bv1C(i)
end do
close(15)

 ! Population D under random mating
marnD=marn
! random mating for n generations
do ijk=1,ngen2
    do i=1,nanim
                  call unif(x1,u)
                  ip=int(u*nmac)+1
                  call unif(x1,u)
                  im=int(u*nhem)+nmac+1
!   father's chromosome
                  do j=1,ncro
                  call unif(x1,u)
                  if (u.lt.0.5) then
                      iale=1
                  else
                      iale=2
                  endif
                  grupoD(i,j,1,1)=marnD(ip,j,1,iale)
                  lam=1.
                  nrec=pois(lam,x1)
                  do ik=1,nrec
```

```
                call unif(x1,u)
                irec(ik)=int(u*(nmar-1))+1
                enddo
                irecord=0
                do ik=1,nrec
                xmin=999999
                    do ij=1,nrec
                     if (irec(ij).lt.xmin) then
                xmin=irec(ij)
                        imin=ij
                     endif
                enddo
                     irecord(ik)=xmin
                     irec(imin)=999999
                enddo
                irecord(nrec+1)=nmar
                if (nrec.eq.0) then
                    do k=2,nmar
                      grupoD(i,j,k,1)=marnD(ip,j,k,iale)
                    enddo
                else
                     ic=2
                     do ik=1,nrec+1
                        do k=ic,irecord(ik)
                                grupoD(i,j,k,1)=marnD(ip,j,k,iale)
                        enddo
                        ic=irecord(ik)+1
                        if (iale.eq.1) then
                                iale=2
                        else
                                iale=1
                        endif
                     enddo
                endif
! mutation
                lam=1.
                nmut=pois(lam,x1)
                do ik=1,nmut
                   call unif(x1,u)
                   imut=int(u*nmar)+1
                    if (grupoD(i,j,imut,1).eq.1) then
                       grupoD(i,j,imut,1)=2
                    else
                       grupoD(i,j,imut,1)=1
                    endif
                 enddo
        enddo
! mother's chromosome
                do j=1,ncro
                call unif(x1,u)
                if (u.lt.0.5) then
                   iale=1
                else
                   iale=2
                endif
                grupoD(i,j,1,2)=marnD(im,j,1,iale)
                lam=1.
                nrec=pois(lam,x1)
                do ik=1,nrec
                call unif(x1,u)
                irec(ik)=int(u*(nmar-1))+1
                enddo
                irecord=0
                do ik=1,nrec
                xmin=999999
                    do ij=1,nrec
                     if (irec(ij).lt.xmin) then
                xmin=irec(ij)
                        imin=ij
                     endif
                enddo
                     irecord(ik)=xmin
                     irec(imin)=999999
                enddo
                irecord(nrec+1)=nmar
                if (nrec.eq.0) then
                  do k=2,nmar
                     grupoD(i,j,k,2)=marnD(im,j,k,iale)
                  enddo
                else
                     ic=2
                    do ik=1,nrec+1
                       do k=ic,irecord(ik)
                         grupoD(i,j,k,2)=marnD(im,j,k,iale)
                       enddo
                       ic=irecord(ik)+1
                        if (iale.eq.1) then
                           iale=2
                       else
                           iale=1
                       endif
                    enddo
        endif
! mutation
        lam=1.
        nmut=pois(lam,x1)
        do ik=1,nmut
        call unif(x1,u)
                imut=int(u*nmar)+1
                if (grupoD(i,j,imut,2).eq.1) then
                    grupoD(i,j,imut,2)=2
                else
                    grupoD(i,j,imut,2)=1
                endif
                enddo
     enddo
   enddo
   marnD=grupoD
print*, ijk
end do

! Final population D
do i=1,nanim2
   call unif(x1,u)
   ip=int(u*nmac)+1
   call unif(x1,u)
   im=int(u*nhem)+nmac+1
!  father's chromosome
   do j=1,ncro
      call unif(x1,u)
      if (u.lt.0.5) then
```

```
          iale=1
        else
          iale=2
        endif
        finalD(i,j,1,1)=grupoD(ip,j,1,iale)
         lam=1.
         nrec=pois(lam,x1)
         do ik=1,nrec
           call unif(x1,u)
irec(ik)=int(u*(nmar-1))+1
enddo
        irecord=0
        do ik=1,nrec
         xmin=999999
                      do ij=1,nrec
                       if (irec(ij).lt.xmin) then
                      xmin=irec(ij)
                          imin=ij
                       endif
enddo
            irecord(ik)=xmin
            irec(imin)=999999
         enddo
         irecord(nrec+1)=nmar
         if (nrec.eq.0) then
                       do k=2,nmar
                       finalD(i,j,k,1)=grupoD(ip,j,k,iale)
                       enddo
          else
                       ic=2
                       do ik=1,nrec+1
                          do k=ic,irecord(ik)
                            finalD(i,j,k,1)=grupoD(ip,j,k,iale)
                          enddo
                          ic=irecord(ik)+1
                          if (iale.eq.1) then
                              iale=2
                          else
                              iale=1
                          endif
                       enddo
         endif
!    mutation
         lam=1.
         nmut=pois(lam,x1)
         do ik=1,nmut
                       call unif(x1,u)
                       imut=int(u*nmar)+1
                       if (finalD(i,j,imut,1).eq.1) then
                           finalD(i,j,imut,1)=2
                       else
                           finalD(i,j,imut,1)=1
                       endif
                      enddo
        enddo
!  mother's chromosome
        do j=1,ncro
        call unif(x1,u)
        if (u.lt.0.5) then
                        iale=1
        else
                        iale=2
        endif
        finalD(i,j,1,2)=grupoD(im,j,1,iale)
         lam=1.
         nrec=pois(lam,x1)
                       do ik=1,nrec
                         call unif(x1,u)
                      irec(ik)=int(u*(nmar-1))+1
                      enddo
                       irecord=0
                       do ik=1,nrec
                        xmin=999999
                          do ij=1,nrec
                          if (irec(ij).lt.xmin) then
                      xmin=irec(ij)
                             imin=ij
                          endif
                      enddo
                          irecord(ik)=xmin
                          irec(imin)=999999
                       enddo
                       irecord(nrec+1)=nmar
                       if (nrec.eq.0) then
                         do k=2,nmar
                          finalD(i,j,k,2)=grupoD(im,j,k,iale)
                         enddo
                       else
                         ic=2
                         do ik=1,nrec+1
                            do k=ic,irecord(ik)
                             finalD(i,j,k,2)=grupoD(im,j,k,iale)
                            enddo
                             ic=irecord(ik)+1
                             if (iale.eq.1) then
                                 iale=2
                             else
                                 iale=1
                             endif
                          enddo
          endif
! mutation
                       lam=1.
                       nmut=pois(lam,x1)
                       do ik=1,nmut
                        call unif(x1,u)
                        imut=int(u*nmar)+1
           if (finalD(i,j,imut,2).eq.1) then
                           finalD(i,j,imut,2)=2
                       else
                 finalD(i,j,imut,2)=1
                        endif
                       enddo
      enddo
enddo
! frequencies of the causative mutations of trait 1
open(16,file='freq1-D.txt')
do i=1,ngenes
p1D(iefec1(i,1),iefec1(i,2))=0
    do k=1,nanim2
     p1D(iefec1(i,1),iefec1(i,2))=p1D(iefec1(i,1),iefec1(i,2))+real(finalD(k,iefec1(i,1),iefec1(i,2),1)+finalD(k,iefec1(i,1),iefec1(i,2),2)-2)
```

```
    end do
    p1D(iefec1(i,1),iefec1(i,2))=p1D(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1D(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance trait 1
Va1=0
var1Df(i)=0
do i=1,ngenes
    var1Df(i)=2*p1D(iefec1(i,1),iefec1(i,2))*(1-p1D(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
     Va1=Va1+var1Df(i)
end do
print *, 'Variance trait 1 population D=',Va1
write(11,*) 'Variance trait 1 population D=',Va1
! frecuencies of causative mutations for trait 2
open(16,file='freq2-D.txt')
do i=1,ngenes
p2D(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
      p2D(iefec2(i,1),iefec2(i,2))=p2D(iefec2(i,1),iefec2(i,2))+real(finalD(k,iefec2(i,1),iefec2(i,2),1)+finalD(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2D(iefec2(i,1),iefec2(i,2))=p2D(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2D(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance trait 2
Va2=0
var2Df(i)=0
do i=1,ngenes
   var2Df(i)=2*p2D(iefec2(i,1),iefec2(i,2))*(1-p2D(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2Df(i)
end do
print *, 'Variance trait 2 population D=',Va2
write(11,*) 'Variance trait 2 population D=',Va2
! simulation of phenotypes
open(15,file='f1D.txt')
open(16,file='f2D.txt')
do i=1,nanim2
yfen1D(i)=100
   do j=1,ngenes
   yfen1D(i)=yfen1D(i)+(finalD(i,iefec1(j,1),iefec1(j,2),1)+finalD(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
   end do
   call normal(x1,u)
   yfen1D(i)=yfen1D(i)+u*sqrt(Vr1)
   write(15,*) yfen1D(i),i,1
end do
close(15)
do i=1,nanim2
yfen2D(i)=1000
   do j=1,ngenes
     yfen2D(i)=yfen2D(i)+(finalD(i,iefec2(j,1),iefec2(j,2),1)+finalD(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
     end do
     call normal(x1,u)
     yfen2D(i)=yfen2D(i)+u*sqrt(Vr2)
     write(16,*) yfen2D(i),i,1
end do
close(16)
! write genotypes
open(14,file='gD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                write(14,*) i,((j-1)*nmar+k),finalD(i,j,k,1),finalD(i,j,k,2)
      end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pD.txt')
do i=1,ncro
   do j=1,nmar
   pD=0
      do k=1,nanim2
                pD=pD+real(finalD(k,i,j,1)+finalD(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pD/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                do g=1,ngenes
                if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                   goto 28
                end if
                if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                   goto 28
                end if
                end do
                write(13,*) i,(j-1)*nmar+k,(finalD(i,j,k,1)+finalD(i,j,k,2)-3)
28    end do
   end do
end do
close(13)
! write neutral marke genotypes in format
open(20,file='nD.txt')
open(13,file='genotD.txt')
do i=1,nanim2
marca(j)=0
   do j=1,neu
   read(20,*) ip,im,iv
   marca(j)=iv+1
   end do
   write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)
! true breeding values for traits1 & 2
open(14,file='tbv2D.txt')
open(15,file='tbv1D.txt')
do i=1,nanim2
bv2D(i)=0
   do j=1,ngenes
     bv2D(i)=bv2D(i)+(finalD(i,iefec2(j,1),iefec2(j,2),1)+finalD(i,iefec2(j,1),iefec2(j,2),2)-3)*efec2(j)
    end do
    write(14,*) i,bv2D(i)
end do
close(14)
```

```
do i=1,nanim2
bv1D(i)=0
   do j=1,ngenes
     bv1D(i)=bv1D(i)+(finalD(i,iefec1(j,1),iefec1(j,2),1)+finalD(i,iefec1(j,1),iefec1(j,2),2)-3)*efec1(j)
   end do
    write(15,*) i,bv1D(i)
end do
close(15)

! Population A+B
open(12,file='f1AB.txt')
open(13,file='f2AB.txt')
do i=1,nanim2
 call unif(x1,u)
 ip=int(nanim2*u)+1
 call unif(x1,u)
 if (u.lt.0.5) then
 yfen1AB(i)=yfen1A(ip)
 yfen2AB(i)=yfen2A(ip)
 do j=1,ncro
    do k=1,nmar
    finalAB(i,j,k,1)=finalA(ip,j,k,1)
    finalAB(i,j,k,2)=finalA(ip,j,k,2)
    end do
 end do
 else
 yfen1AB(i)=yfen1B(ip)
 yfen2AB(i)=yfen2B(ip)
 do j=1,ncro
    do k=1,nmar
    finalAB(i,j,k,1)=finalB(ip,j,k,1)
    finalAB(i,j,k,2)=finalB(ip,j,k,2)
    end do
 end do
 end if
 write(12,*) yfen1AB(i),i,1
 write(13,*) yfen2AB(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations of trait 1
open(16,file='freq1-AB.txt')
do i=1,ngenes
p1AB(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
    p1AB(iefec1(i,1),iefec1(i,2))=p1AB(iefec1(i,1),iefec1(i,2))+real(finalAB(k,iefec1(i,1),iefec1(i,2),1)+finalAB(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
    p1AB(iefec1(i,1),iefec1(i,2))=p1AB(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1AB(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1ABf(i)=0
do i=1,ngenes
     var1ABf(i)=2*p1AB(iefec1(i,1),iefec1(i,2))*(1-  p1AB(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
     Va1=Va1+var1ABf(i)
end do
print *, 'Variance trait 1 population A+B=',Va1
write(11,*) 'Variance trait 1 population A+B=',Va1
! frequencies of the causative mutations of trait 2
open(16,file='freq2-AB.txt')
do i=1,ngenes
p2AB(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
   p2AB(iefec2(i,1),iefec2(i,2))=p2AB(iefec2(i,1),iefec2(i,2))+real(finalAB(k,iefec2(i,1),iefec2(i,2),1)+finalAB(k,iefec2(i,1),iefec2(i,2),2)-2)
   end do
   p2AB(iefec2(i,1),iefec2(i,2))=p2AB(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2AB(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance trait 2
Va2=0
do i=1,ngenes
   var2ABf(i)=2*p2AB(iefec2(i,1),iefec2(i,2))*(1-p2AB(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2ABf(i)
end do
print *, 'Variance trait 2 population A+B=',Va2
write(11,*) 'Variance trait 2 population A+B=',Va2
! write genotypes
open(14,file='gAB.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                  write(14,*) i,((j-1)*nmar+k),finalAB(i,j,k,1),finalAB(i,j,k,2)
      end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pAB.txt')
do i=1,ncro
   do j=1,nmar
   pAB=0
      do k=1,nanim2
       pAB=pAB+real(finalAB(k,i,j,1)+finalAB(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pAB/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nAB.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
        do g=1,ngenes
                  if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 29
                  end if
                  if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 29
                  end if
        end do
        write(13,*) i,(j-1)*nmar+k,(finalAB(i,j,k,1)+finalAB(i,j,k,2)-3)
29   end do
   end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nAB.txt')
```

```
open(13,file='genotAB.txt')
do i=1,nanim2
  marca(j)=0
  do j=1,neu
  read(20,*) ip,im,iv
  marca(j)=iv+1
  end do
  write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)

! Population A+C
open(12,file='f1AC.txt')
open(13,file='f2AC.txt')
do i=1,nanim2
  call unif(x1,u)
  ip=int(nanim2*u)+1
  call unif(x1,u)
  if (u.lt.0.5) then
    yfen1AC(i)=yfen1A(ip)
    yfen2AC(i)=yfen2A(ip)
    do j=1,ncro
      do k=1,nmar
                    finalAC(i,j,k,1)=finalA(ip,j,k,1)
                    finalAC(i,j,k,2)=finalA(ip,j,k,2)
      end do
    end do
  else
    yfen1AC(i)=yfen1C(ip)
    yfen2AC(i)=yfen2C(ip)
    do j=1,ncro
      do k=1,nmar
      finalAC(i,j,k,1)=finalC(ip,j,k,1)
      finalAC(i,j,k,2)=finalC(ip,j,k,2)
      end do
    end do
  end if
  write(12,*) yfen1AC(i),i,1
  write(13,*) yfen2AC(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations for trait 1
open(16,file='freq1-AC.txt')
do i=1,ngenes
p1AC(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
    p1AC(iefec1(i,1),iefec1(i,2))=p1AC(iefec1(i,1),iefec1(i,2))+real(finalAC(k,iefec1(i,1),iefec1(i,2),1)+finalAC(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
   p1AC(iefec1(i,1),iefec1(i,2))=p1AC(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1AC(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1ACf(i)=0
do i=1,ngenes
   var1ACf(i)=2*p1AC(iefec1(i,1),iefec1(i,2))*(1-p1AC(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
   Va1=Va1+var1ACf(i)
end do
print *, 'Variance trait 1 population A+C=',Va1
write(11,*) 'Variance trait 1 population A+C=',Va1
! frequencies of the causative mutations for trait 2
open(16,file='freq2-AC.txt')
do i=1,ngenes
p2AC(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
    p2AC(iefec2(i,1),iefec2(i,2))=p2AC(iefec2(i,1),iefec2(i,2))+real(finalAC(k,iefec2(i,1),iefec2(i,2),1)+finalAC(k,iefec2(i,1),iefec2(i,2),2)-2)
   end do
   p2AC(iefec2(i,1),iefec2(i,2))=p2AC(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2AC(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance of trait 2
Va2=0
var2ACf(i)=0
do i=1,ngenes
    var2ACf(i)=2*p2AC(iefec2(i,1),iefec2(i,2))*(1-p2AC(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
    Va2=Va2+var2ACf(i)
end do
print *, 'Variance trait 2 population A+C=',Va2
write(11,*) 'Variance trait 2 population A+C=',Va2
! write genotypes
open(14,file='gAC.txt')
do i=1,nanim2
   do j=1,ncro
     do k=1,nmar
                    write(14,*) i,((j-1)*nmar+k),finalAC(i,j,k,1),finalAC(i,j,k,2)
     end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pAC.txt')
do i=1,ncro
   do j=1,nmar
   pAC=0
      do k=1,nanim2
                    pAC=pAC+real(finalAC(k,i,j,1)+finalAC(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pAC/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nAC.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                    do g=1,ngenes
                    if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                      goto 30
                    end if
                    if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                      goto 30
                    end if
                    end do
                    write(13,*) i,(j-1)*nmar+k,(finalAC(i,j,k,1)+finalAC(i,j,k,2)-3)
30    end do
   end do
```

```
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nAC.txt')
open(13,file='genotAC.txt')
do i=1,nanim2
  marca(j)=0
  do j=1,neu
  read(20,*) ip,im,iv
  marca(j)=iv+1
  end do
  write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)

! Population A+D
open(12,file='f1AD.txt')
open(13,file='f2AD.txt')
do i=1,nanim2
  call unif(x1,u)
  ip=int(nanim2*u)+1
  call unif(x1,u)
  if (u.lt.0.5) then
     yfen1AD(i)=yfen1A(ip)
     yfen2AD(i)=yfen2A(ip)
     do j=1,ncro
        do k=1,nmar
                        finalAD(i,j,k,1)=finalA(ip,j,k,1)
                        finalAD(i,j,k,2)=finalA(ip,j,k,2)
        end do
     end do
  else
     yfen1AD(i)=yfen1D(ip)
     yfen2AD(i)=yfen2D(ip)
     do j=1,ncro
        do k=1,nmar
                        finalAD(i,j,k,1)=finalD(ip,j,k,1)
                        finalAD(i,j,k,2)=finalD(ip,j,k,2)
         end do
     end do
  end if
  write(12,*) yfen1AD(i),i,1
  write(13,*) yfen2AD(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations for trait 1
open(16,file='freq1-AD.txt')
do i=1,ngenes
p1AD(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
      p1AD(iefec1(i,1),iefec1(i,2))=p1AD(iefec1(i,1),iefec1(i,2))+real(finalAD(k,iefec1(i,1),iefec1(i,2),1)+finalAD(k,iefec1(i,1),iefec1(i,2),2)-2)
    end do
    p1AD(iefec1(i,1),iefec1(i,2))=p1AD(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1AD(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1ADf(i)=0
do i=1,ngenes
   var1ADf(i)=2*p1AD(iefec1(i,1),iefec1(i,2))*(1-p1AD(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
   Va1=Va1+var1ADf(i)
end do
print *, 'Variance trait 1 population A+D=',Va1
write(11,*) 'Variance trait 1 population A+D=',Va1
! frequencies of the causative mutations for trair 2
open(16,file='freq2-AD.txt')
do i=1,ngenes
p2AD(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
      p2AD(iefec2(i,1),iefec2(i,2))=p2AD(iefec2(i,1),iefec2(i,2))+real(finalAD(k,iefec2(i,1),iefec2(i,2),1)+finalAD(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2AD(iefec2(i,1),iefec2(i,2))=p2AD(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2AD(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance of trait 2
Va2=0
var2ADf(i)=0
do i=1,ngenes
   var2ADf(i)=2*p2AD(iefec2(i,1),iefec2(i,2))*(1-p2AD(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2ADf(i)
end do
print *, 'Variance trait 2 population A+D=',Va2
write(11,*) 'Variance trait 2 population A+D=',Va2
! write genotypes
open(14,file='gAD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                        write(14,*) i,((j-1)*nmar+k),finalAD(i,j,k,1),finalAD(i,j,k,2)
      end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pAD.txt')
do i=1,ncro
   do j=1,nmar
    pAD=0
      do k=1,nanim2
                        pAD=pAD+real(finalAD(k,i,j,1)+finalAD(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pAD/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nAD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
        do g=1,ngenes
                        if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                          goto 31
                        end if
                        if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                          goto 31
                        end if
```

```
          end do
                    write(13,*) i,(j-1)*nmar+k,(finalAD(i,j,k,1)+finalAD(i,j,k,2)-3)
31     end do
     end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nAD.txt')
open(13,file='genotAD.txt')
do i=1,nanim2
  marca(j)=0
  do j=1,neu
  read(20,*) ip,im,iv
  marca(j)=iv+1
  end do
  write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)

! Population B+C
open(12,file='f1BC.txt')
open(13,file='f2BC.txt')
do i=1,nanim2
  call unif(x1,u)
  ip=int(nanim2*u)+1
  call unif(x1,u)
  if (u.lt.0.5) then
     yfen1BC(i)=yfen1B(ip)
     yfen2BC(i)=yfen2B(ip)
     do j=1,ncro
        do k=1,nmar
                    finalBC(i,j,k,1)=finalB(ip,j,k,1)
                    finalBC(i,j,k,2)=finalB(ip,j,k,2)
        end do
     end do
  else
     yfen1BC(i)=yfen1C(ip)
     yfen2BC(i)=yfen2C(ip)
     do j=1,ncro
        do k=1,nmar
                    finalBC(i,j,k,1)=finalC(ip,j,k,1)
                    finalBC(i,j,k,2)=finalC(ip,j,k,2)
        end do
     end do
  end if
  write(12,*) yfen1BC(i),i,1
  write(13,*) yfen2BC(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations for trait 1
open(16,file='freq1-BC.txt')
do i=1,ngenes
p1BC(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
     p1BC(iefec1(i,1),iefec1(i,2))=p1BC(iefec1(i,1),iefec1(i,2))+real(finalBC(k,iefec1(i,1),iefec1(i,2),1)+finalBC(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
   p1BC(iefec1(i,1),iefec1(i,2))=p1BC(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1BC(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1BCf(i)=0
do i=1,ngenes
   var1BCf(i)=2*p1BC(iefec1(i,1),iefec1(i,2))*(1-p1BC(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
   Va1=Va1+var1BCf(i)
end do
print *, 'Variance trait 1 population B+C=',Va1
write(11,*) 'Variance trait 1 population B+C=',Va1
! frequencies of the causative mutations for trait 2
open(16,file='freq2-BC.txt')
do i=1,ngenes
p2BC(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
     p2BC(iefec2(i,1),iefec2(i,2))=p2BC(iefec2(i,1),iefec2(i,2))+real(finalBC(k,iefec2(i,1),iefec2(i,2),1)+finalBC(k,iefec2(i,1),iefec2(i,2),2)-2)
   end do
   p2BC(iefec2(i,1),iefec2(i,2))=p2BC(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2BC(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance of trait 2
Va2=0
var2BCf(i)=0
do i=1,ngenes
     var2BCf(i)=2*p2BC(iefec2(i,1),iefec2(i,2))*(1-p2BC(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
     Va2=Va2+var2BCf(i)
end do
print *, 'Variance trait 2 population B+C=',Va2
write(11,*) 'Variance trait 2 population B+C=',Va2
! write genotype
open(14,file='gBC.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                    write(14,*) i,((j-1)*nmar+k),finalBC(i,j,k,1),finalBC(i,j,k,2)
      end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pBC.txt')
do i=1,ncro
   do j=1,nmar
   pBC=0
     do k=1,nanim2
                    pBC=pBC+real(finalBC(k,i,j,1)+finalBC(k,i,j,2)-2)
     end do
     write(13,*) (i-1)*nmar+j,pBC/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nBC.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                    do g=1,ngenes
                    if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                      goto 32
```

```
                    end if
                    if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                      goto 32
                    end if
                    end do
                    write(13,*) i,(j-1)*nmar+k,(finalBC(i,j,k,1)+finalBC(i,j,k,2)-3)
32    end do
   end do
end do
close(13)
! write neutral markers genotypes in format
open(20,file='nBC.txt')
open(13,file='genotBC.txt')
do i=1,nanim2
  marca(j)=0
  do j=1,neu
  read(20,*) ip,im,iv
  marca(j)=iv+1
  end do
  write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)

! Population B+D
open(12,file='f1BD.txt')
open(13,file='f2BD.txt')
do i=1,nanim2
  call unif(x1,u)
  ip=int(nanim2*u)+1
  call unif(x1,u)
  if (u.lt.0.5) then
     yfen1BD(i)=yfen1B(ip)
     yfen2BD(i)=yfen2B(ip)
     do j=1,ncro
       do k=1,nmar
                    finalBD(i,j,k,1)=finalB(ip,j,k,1)
                    finalBD(i,j,k,2)=finalB(ip,j,k,2)
       end do
     end do
  else
     yfen1BD(i)=yfen1D(ip)
     yfen2BD(i)=yfen2D(ip)
     do j=1,ncro
       do k=1,nmar
                    finalBD(i,j,k,1)=finalD(ip,j,k,1)
                    finalBD(i,j,k,2)=finalD(ip,j,k,2)
       end do
     end do
  end if
  write(12,*) yfen1BD(i),i,1
  write(13,*) yfen2BD(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations of trait 1
open(16,file='freq1-BD.txt')
do i=1,ngenes
p1BD(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
    p1BD(iefec1(i,1),iefec1(i,2))=p1BD(iefec1(i,1),iefec1(i,2))+real(finalBD(k,iefec1(i,1),iefec1(i,2),1)+finalBD(k,iefec1(i,1),iefec1(i,2),2)-2)
    end do
    p1BD(iefec1(i,1),iefec1(i,2))=p1BD(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1BD(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1BDf(i)=0
do i=1,ngenes
   var1BDf(i)=2*p1BD(iefec1(i,1),iefec1(i,2))*(1-p1BD(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
   Va1=Va1+var1BDf(i)
end do
print *, 'Variance trait 1 population B+D=',Va1
write(11,*) 'Variance trait 1 population B+D=',Va1
! frequencies of the causative mutations of trait 2
open(16,file='freq2-BD.txt')
do i=1,ngenes
p2BD(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2
      p2BD(iefec2(i,1),iefec2(i,2))=p2BD(iefec2(i,1),iefec2(i,2))+real(finalBD(k,iefec2(i,1),iefec2(i,2),1)+finalBD(k,iefec2(i,1),iefec2(i,2),2)-2)
   end do
   p2BD(iefec2(i,1),iefec2(i,2))=p2BD(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2BD(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance of trait 2
Va2=0
var2BDf(i)=0
do i=1,ngenes
   var2BDf(i)=2*p2BD(iefec2(i,1),iefec2(i,2))*(1-p2BD(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2BDf(i)
end do
print *, 'Variance trait 2 population B+D=',Va2
write(11,*) 'Variance trait 2 population B+D=',Va2
! write genotypes
open(14,file='gBD.txt')
do i=1,nanim2
   do j=1,ncro
     do k=1,nmar
                    write(14,*) i,((j-1)*nmar+k),finalBD(i,j,k,1),finalBD(i,j,k,2)
     end do
   end do
end do
close(14)
! marker and gene frequencies
open(13,file='pBD.txt')
do i=1,ncro
   do j=1,nmar
   pBD=0
     do k=1,nanim2
                    pBD=pBD+real(finalBD(k,i,j,1)+finalBD(k,i,j,2)-2)
     end do
     write(13,*) (i-1)*nmar+j,pBD/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nBD.txt')
do i=1,nanim2
   do j=1,ncro
```

```
      do k=1,nmar
                  do g=1,ngenes
                  if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 33
                  end if
                  if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 33
                  end if
                  end do
                  write(13,*) i,(j-1)*nmar+k,(finalBD(i,j,k,1)+finalBD(i,j,k,2)-3)
33    end do
    end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nBD.txt')
open(13,file='genotBD.txt')
do i=1,nanim2
  marca(j)=0
  do j=1,neu
  read(20,*) ip,im,iv
  marca(j)=iv+1
  end do
  write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)

! Population C+D
open(12,file='f1CD.txt')
open(13,file='f2CD.txt')
do i=1,nanim2
  call unif(x1,u)
  ip=int(nanim2*u)+1
  call unif(x1,u)
  if (u.lt.0.5) then
     yfen1CD(i)=yfen1C(ip)
     yfen2CD(i)=yfen2C(ip)
     do j=1,ncro
        do k=1,nmar
                  finalCD(i,j,k,1)=finalC(ip,j,k,1)
                  finalCD(i,j,k,2)=finalC(ip,j,k,2)
        end do
     end do
  else
     yfen1CD(i)=yfen1D(ip)
     yfen2CD(i)=yfen2D(ip)
     do j=1,ncro
        do k=1,nmar
                  finalCD(i,j,k,1)=finalD(ip,j,k,1)
                  finalCD(i,j,k,2)=finalD(ip,j,k,2)
        end do
     end do
  end if
  write(12,*) yfen1CD(i),i,1
  write(13,*) yfen2CD(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations of trait 1
open(16,file='freq1-CD.txt')
do i=1,ngenes
p1CD(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2
    p1CD(iefec1(i,1),iefec1(i,2))=p1CD(iefec1(i,1),iefec1(i,2))+real(finalCD(k,iefec1(i,1),iefec1(i,2),1)+finalCD(k,iefec1(i,1),iefec1(i,2),2)-2)
    end do
    p1CD(iefec1(i,1),iefec1(i,2))=p1CD(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1CD(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1CDf(i)=0
do i=1,ngenes
    var1CDf(i)=2*p1CD(iefec1(i,1),iefec1(i,2))*(1-p1CD(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
    Va1=Va1+var1CDf(i)
end do
print *, 'Variance trait 1 population C+D=',Va1
write(11,*) 'Variance trait 1 population C+D=',Va1
! frequencies of the causative mutations of trait 2
open(16,file='freq2-CD.txt')
do i=1,ngenes
p2CD(iefec2(i,1),iefec2(i,2))=0
    do k=1,nanim2
     p2CD(iefec2(i,1),iefec2(i,2))=p2CD(iefec2(i,1),iefec2(i,2))+real(finalCD(k,iefec2(i,1),iefec2(i,2),1)+finalCD(k,iefec2(i,1),iefec2(i,2),2)-2)
    end do
    p2CD(iefec2(i,1),iefec2(i,2))=p2CD(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
    write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2CD(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance of trait 2
Va2=0
var2CDf(i)=0
do i=1,ngenes
   var2CDf(i)=2*p2CD(iefec2(i,1),iefec2(i,2))*(1-p2CD(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2CDf(i)
end do
print *, 'Variance trait 2 population C+D=',Va2
write(11,*) 'Variance trait 2 population C+D=',Va2
! write genotypes
open(14,file='gCD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                  write(14,*) i,((j-1)*nmar+k),finalCD(i,j,k,1),finalCD(i,j,k,2)
      end do
    end do
end do
close(14)
! markers and genes frequencies
open(13,file='pCD.txt')
do i=1,ncro
   do j=1,nmar
   pCD=0
      do k=1,nanim2
                  pCD=pCD+real(finalCD(k,i,j,1)+finalCD(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pCD/(2*real(nanim2))
   end do
end do
close(13)
```

```
! write neutral marker genotypes
open(13,file='nCD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                  do g=1,ngenes
                  if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 34
                  end if
                  if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 34
                  end if
                  end do
                  write(13,*) i,(j-1)*nmar+k,(finalCD(i,j,k,1)+finalCD(i,j,k,2)-3)
34    end do
   end do
end do
close(13)
! write neautral marker genotypes in format
open(20,file='nCD.txt')
open(13,file='genotCD.txt')
do i=1,nanim2
 marca(j)=0
 do j=1,neu
 read(20,*) ip,im,iv
 marca(j)=iv+1
 end do
 write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)

! Population A+B+C+D
open(12,file='f1ABCD.txt')
open(13,file='f2ABCD.txt')
do i=1,nanim2
 call unif(x1,u)
 ip=int(nanim2*u)+1
 call unif(x1,u)
 if (u.lt.0.25) then
    yfen1ABCD(i)=yfen1A(ip)
    yfen2ABCD(i)=yfen2A(ip)
    do j=1,ncro
       do k=1,nmar
                  finalABCD(i,j,k,1)=finalA(ip,j,k,1)
                  finalABCD(i,j,k,2)=finalA(ip,j,k,2)
        end do
    end do
 end if
 if (u.gt.0.25.and.u.lt.0.5) then
    yfen1ABCD(i)=yfen1B(ip)
    yfen2ABCD(i)=yfen2B(ip)
    do j=1,ncro
       do k=1,nmar
                  finalABCD(i,j,k,1)=finalB(ip,j,k,1)
                  finalABCD(i,j,k,2)=finalB(ip,j,k,2)
       end do
    end do
 end if
 if (u.gt.0.5.and.u.lt.0.75) then
    yfen1ABCD(i)=yfen1C(ip)
    yfen2ABCD(i)=yfen2C(ip)
    do j=1,ncro
       do k=1,nmar
                  finalABCD(i,j,k,1)=finalC(ip,j,k,1)
                  finalABCD(i,j,k,2)=finalC(ip,j,k,2)
       end do
    end do
 end if
 if (u.gt.0.75) then
    yfen1ABCD(i)=yfen1D(ip)
    yfen2ABCD(i)=yfen2D(ip)
    do j=1,ncro
       do k=1,nmar
                  finalABCD(i,j,k,1)=finalD(ip,j,k,1)
                  finalABCD(i,j,k,2)=finalD(ip,j,k,2)
       end do
    end do
 end if
 write(12,*) yfen1ABCD(i),i,1
 write(13,*) yfen2ABCD(i),i,1
end do
close(12)
close(13)
! frequencies of the causative mutations of trait 1
open(16,file='freq1-ABCD.txt')
do i=1,ngenes
p1ABCD(iefec1(i,1),iefec1(i,2))=0
   do k=1,nanim2   p1ABCD(iefec1(i,1),iefec1(i,2))=p1ABCD(iefec1(i,1),iefec1(i,2))+real(finalABCD(k,iefec1(i,1),iefec1(i,2),1)+finalABCD(k,iefec1(i,1),iefec1(i,2),2)-2)
   end do
   p1ABCD(iefec1(i,1),iefec1(i,2))=p1ABCD(iefec1(i,1),iefec1(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec1(i,1)-1)*nmar+iefec1(i,2)),p1ABCD(iefec1(i,1),iefec1(i,2))
end do
close(16)
! additive variance of trait 1
Va1=0
var1ABCDf(i)=0
do i=1,ngenes
   var1ABCDf(i)=2*p1ABCD(iefec1(i,1),iefec1(i,2))*(1-p1ABCD(iefec1(i,1),iefec1(i,2)))*efec1(i)**2
    Va1=Va1+var1ABCDf(i)
end do
print *, 'Variance trait 1 population A+B+C+D=',Va1
write(11,*) 'Variance trait 1 population A+B+C+D=',Va1
! frequencies of the causative mutations of trait 2
open(16,file='freq2-ABCD.txt')
do i=1,ngenes
p2ABCD(iefec2(i,1),iefec2(i,2))=0
   do k=1,nanim2   p2ABCD(iefec2(i,1),iefec2(i,2))=p2ABCD(iefec2(i,1),iefec2(i,2))+real(finalABCD(k,iefec2(i,1),iefec2(i,2),1)+finalABCD(k,iefec2(i,1),iefec2(i,2),2)-2)
   end do
   p2ABCD(iefec2(i,1),iefec2(i,2))=p2ABCD(iefec2(i,1),iefec2(i,2))/(2*real(nanim2))
   write(16,*) i,((iefec2(i,1)-1)*nmar+iefec2(i,2)),p2ABCD(iefec2(i,1),iefec2(i,2))
end do
close(16)
! additive variance
Va2=0
var2ABCDf(i)=0
do i=1,ngenes
   var2ABCDf(i)=2*p2ABCD(iefec2(i,1),iefec2(i,2))*(1-p2ABCD(iefec2(i,1),iefec2(i,2)))*efec2(i)**2
   Va2=Va2+var2ABCDf(i)
end do
print *, 'Variance trait 2 population A+B+C+D=',Va2
```

```
write(11,*) 'Variance trait 2 population A+B+C+D=',Va2
! write genotypes
open(14,file='gABCD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                  write(14,*) i,((j-1)*nmar+k),finalABCD(i,j,k,1),finalABCD(i,j,k,2)
       end do
    end do
end do
close(14)
! markers and genes frequencies
open(13,file='pABCD.txt')
do i=1,ncro
   do j=1,nmar
   pABCD=0
      do k=1,nanim2
                  pABCD=pABCD+real(finalABCD(k,i,j,1)+finalABCD(k,i,j,2)-2)
      end do
      write(13,*) (i-1)*nmar+j,pABCD/(2*real(nanim2))
   end do
end do
close(13)
! write neutral marker genotypes
open(13,file='nABCD.txt')
do i=1,nanim2
   do j=1,ncro
      do k=1,nmar
                  do g=1,ngenes
                  if (j==iefec1(g,1).and.k==iefec1(g,2)) then
                    goto 35
                  end if
                  if (j==iefec2(g,1).and.k==iefec2(g,2)) then
                    goto 35
                  end if
                  end do
                  write(13,*) i,(j-1)*nmar+k,(finalABCD(i,j,k,1)+finalABCD(i,j,k,2)-3)
35    end do
   end do
end do
close(13)
! write neutral marker genotypes in format
open(20,file='nABCD.txt')
open(13,file='genotABCD.txt')
do i=1,nanim2
  marca(j)=0
  do j=1,neu
  read(20,*) ip,im,iv
marca(j)=iv+1
  !print *, i,marca(j)
end do
  write(13,'(i4,1x,30000i1)') i,(marca(j),j=1,neu)
end do
close(13)
close(20)
close(11)
end

! ------------------------------------------------------------------------------
     subroutine normal(x1,z)
!     generacion de un numero normal z -> n(0,1)
!     x1 es la semilla
     implicit double precision(a-h,o-z)
     real*8 x1,z,u1,u2
     call unif(x1,u1)
     call unif(x1,u2)
     z=((-2.*log(u1))**0.5)*cos(2.*3.1416*u2)
return
end
! ------------------------------------------------------------------------------
     subroutine unif(x1,u)
!     generacion de un numero uniforme u[0,1]
!     x1 es la semilla
     implicit double precision(a-h,o-z)
divis=2.**63.-1.
     trans=7**5
     divid=trans*x1
     lsol=int(divid/divis)
x1=divid-lsol*divis
     u=x1/divis
!u=rand()
return
     end
!------------------------------------------------------------------------------
function pois(lam,x1)
implicit none
integer i,pois,n
logical o
real,parameter:: e=2.71828
real *8 x1,sum,u,lam
real *8 pro(0:100),prosum(0:100)
pro=0
prosum=0
sum=0
do n=0,100
pro(n)=1
   do i=1,n
   pro(n)=pro(n)*i
   end do
   pro(n)=(e**(-lam)*lam**n)/pro(n)
   sum=sum+pro(n)
   prosum(n)=sum
   ! print*, prosum(n)
end do
call unif (x1,u)
 !print *,u
o=.true.
i=0
do while (o)
if (u>=prosum(i)) then
  i=i+1
else
  pois=i
  O=.false.
endif
end do
return
end
```