GS3

GENOMIC SELECTION — GIBBS SAMPLING — GAUSS SEIDEL

(AND BAYESCπ)

Andrés Legarra [1] [2] Anne Ricard [3] [4] Olivier Filangi [5] [6]

October 11, 2011

[1]andres.legarra [at] toulouse.inra.fr
[2]INRA, UR 631, F-31326 Auzeville, France
[3]anne.ricard [at] toulouse.inra.fr
[4]INRA, UMR 1313, 78352 Jouy-en-Josas, France
[5]olivier.filangi [at] rennes.inra.fr
[6]INRA, UMR 598 35042 Rennes, France

This program has been partially financed by FEDER European funds through POCTEFA: http://www.poctefa.eu/.

COOPERACIÓN COOPÉRATION TERRITORIAL TERRITORIALE

ESPAÑA-FRANCE-ANDORRA

2007-2013

Unión Europea

Fondo Europeo de Desarrollo Regional

Contents

1 Introduction

This draft describes using and understanding a software for genome-wide genetic evaluations and validations, inspired in the theory by [11], and used for own our research in [9].

In short: it estimates effects of SNPs, either using a priori normal distributions (GBLUP), or the Bayesian Lasso [13, 1, 8] or a mixture of π normal and $1-\pi$ a mass point at 0, namely BayesC(Pi) [5, 2]. *Note that our definition of π here is* opposite *to those authors: π = the fraction of SNPs "having" an effect.*

The program is self-contained, using modules from Ignacy Misztal's BLUPF90 distribution at `http://nce.ads.uga.edu/~ignacy`. Some functions and subroutines have been taken from the Alan Miller web page at `http://users.bigpond.net.au/amiller/`. It has been tested with NAG f95, ifort and gfortran >= 4.3. Gustavo de los Campos helped us with the heterogenous variances and an R code for the Bayesian Lasso.

The computing methods have been described in [7], as well as in [2].

1.1 History

We wrote this program to implement genome-wide genetic evaluation (*aka* genomic selection) in mice [9], as there was nothing available around. The program uses Gibbs sampling, by means of an unconventional Gibbs sampling scheme [7]. It accepts quite general models.

We added BayesCPi end 2010, motivated basically for GWAS; and Bayesian Lasso in August 2011 as our previous version was not very user-friendly.

2 Background

Recently, the availability of massive "cheap" marker genotyping raised up the question on how to use these data for genetic evaluation and marker assisted selection. Proposals by [6, 11] among others, use a linear model for this purpose, in which each marker variant across the genome is assigned a linear effect, as follows:

$$y_i = \sum_{j=1}^{n} (z_{ijk} a_{jk}) + e_i$$

where y_i is the phenotype of the i-th animal, z_{ijk} is an indicator covariate for the i-th animal and the j-th marker locus in its k-th allelic form, and e_i

is a residual term. Hereinafter and for the sake of clarity we will refer to a_{jk} as *"marker locus effects"*.

For the sake of simplicity, we further assumed biallelic loci and a simpler model as follows. In the j-th locus, there are two possible alleles for each SNP (say 1 and 2), and there are three possible genotypes: "11", "12" and "22". We arbitrarily assign the value $+\frac{1}{2}a_j$ to the allele 1 and the value $-\frac{1}{2}a_j$ to the allele 2. This follows a classical parameterization in which a_j is half the difference between the two homozygotes [10]. These are the additive effects of the SNP's and they can be thought of as classical substitution effects in the infinitesimal model.

As for the dominant effect d_j, it comes up when the genotype is "12".

3 Models

3.1 General model

The following kind of linear models is supported:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Za} + \mathbf{Wd} + \mathbf{Tu} + \mathbf{Sc} + \mathbf{e} \tag{1}$$

Including any number and kind (cross-classified, covariates) fixed effects ($\mathbf{b}$), and random (multivariate normal) additive $\mathbf{a}$ and dominant $\mathbf{d}$ marker locus effects, polygenic infinitesimal effects $\mathbf{u}$, and random environmental effects $\mathbf{c}$.

If the prior distribution of $\mathbf{a}$ is considered to be normal [14], this model is often called GBLUP. Random effects have associated variance components. You can estimate them using the software, or (much faster), if you have previous estimates of genetic variance σ_u^2, you can use an *approximate* formula which is extensively discussed in [3]: $\sigma_a^2 = \sigma_u^2/2\sum p_i q_i$ where p_i is the allelic frequency at SNP i.

3.2 Heterogeneity of variances

Heterogeneity of variances in the residual is accepted (v.gr., for use of DYD's with their accuracies) through a column of weights. These works as follows: let ω_i be the weight for record i. These implies that the distribution for y_i is:

$y_i|\cdots = N(\hat{y}_i, \sigma_e^2/\omega_i)$, where $\hat{y}_i = \mathbf{x}_i\mathbf{b} + \mathbf{z}_i\mathbf{a} + \mathbf{w}_i\mathbf{d} + \mathbf{t}_i\mathbf{u} + \mathbf{s}_i\mathbf{c}$.

Thus $\mathbf{e} \sim N(0, \mathbf{R})$, where $\mathbf{R}_{i,i} = \sigma_e^2/\omega_i$.

In a typical case, weights ω are reliabilities of DYD's expressed as "equivalent daughter contributions".

3.3 Submodels

Any submodel from the above can be used *but* random effects can only be included once, e.g., there is no possibility of including two random environmental effects (say litter and herd-year-season).

3.4 Mixture (BayesCPi) modelling of marker locus effects

It is reasonable to assume that most marker loci are *not* in linkage disequilibrium with markers. A way of selecting a subset of them is by fixing a non-negligible *a priori* probability of their effects to be zero. Method BayesB [11] achieved this through variance components having values of zero. An alternative approach is to set up an indicator variable (δ) stating whether the marker has any effect (1) or not (0). That is, the model becomes:

$$y_i = \text{other effects} + \sum_{j=1}^{n} (z_{ij} a_j \delta_j) + e_i$$

with $\delta_j = (0, 1)$. The distribution of $\boldsymbol{\delta} = (\delta_1 \ldots \delta_n)$ can be posited as a binomial, with probability π. This model (a mixture model) is more parsimonious than [11] and MCMC is straightforward [2]. On the other hand a prior distribution has to be postulated for π, and this is a beta distribution. Details can be found in [5].

3.5 Bayesian Lasso

The Lasso (least absolute shrinkage and selection operator [13]) combines variable selection and shrinkage. Its Bayesian counterpart, the Bayesian Lasso [12] provides a more natural interpretation in terms of a priori distributions. In particular, Bayesian Lasso provides a fully parametric model with a simple Gibbs sampler implementation. Further, the exponential distribution of the Lasso is thought to reflect reasonably well the nature of quantitative trait locus (QTL) effects [4]. The Bayesian Lasso has been used in genomic selection with good results [1, 8]. There are two possible implementations of the Bayesian Lasso [13, 12]; [8] compared both. In this program, only Tibshirani's implementation is used; this was called BL2Var by [8]. To use Park & Casella 's [12], I recommend package BLR for R, available in http://cran.r-project.org/web/packages/BLR/index.html.

For an individual SNP, the prior distribution is thus as follows:

$$\Pr(a_i|\lambda) = \frac{\lambda}{2}\exp(-\lambda|a_i|)$$

But this can be written as:

$$\Pr(a_i|\tau^2) = N(0, \tau_i^2)$$

$$\Pr(\tau_i^2) = \frac{\lambda^2}{2}\exp(-\lambda^2|\tau_i^2|)$$

So, basically we are estimating individual variances fo each SNP (as in BayesB). These variances can be used to weight each SNP when constructing a genomic relationship matrix.

3.6 A priori information

Prior inverted-chi squared distributions can be postulated for variance components σ_a^2, σ_d^2, σ_u^2, σ_c^2, σ_e^2 for estimation with `VCE`. These are also starting values. For ease of use, we have considered that beta distributions (with α and β parameters) for π and inverted-chi squared distributions for the different variances. Note that values of $\alpha = 0$ or $\beta = 0$ will cause problems because the Beta distribution will be ill-defined. Note also that

- $\alpha = 1$, $\beta = 1 \rightarrow$ uniform distribution on π.
- $\alpha = 1$, $\beta = 10d10 \rightarrow \pi$ almost certainly close to 0 (most SNPs have no effect).
- $\alpha = 10d8$, $\beta = 10d10 \rightarrow \pi$ almost exactly fixed to 0.01 (on average, 10% SNPs will have an effect).

These prior distributions are used when a full MCMC is run but not for `BLUP` estimation or in the `PREDICT` option.

For λ the prior is bounded between 0 and 10^7.

4 Functionality

4.1 MCMC

A full MCMC is run with the keyword `VCE`. This samples all possible unknowns $(\mathbf{y}, \mathbf{b}, \mathbf{a}, \mathbf{d}, \mathbf{u}, \mathbf{c}, \sigma_a^2, \sigma_d^2, \sigma_u^2, \sigma_c^2, \sigma_e^2)$ and $\boldsymbol{\delta}$ and hyperparameter π if requested . Output are samples of variance components components and π and a posteriori means for $\mathbf{b}, \mathbf{a}, \mathbf{d}, \mathbf{u}, \mathbf{c}$. "Generalized" genomic breeding value estimates (EBV's) are also in the output.

Continuation (in the case of sudden interruption or just the desire of running more iterations) are possible via a specific keyword (but not for the Bayesian Lasso). The continuation is done by reading the last saved state of the MCMC chain, so be careful not to delete that file (named `parameter_file_cont`).

4.2 BLUP

BLUP is defined here in the spirit of Henderson's BLUP, as in [11]. Therefore it is an estimator that assumes known variances for all random effects and $\boldsymbol{\delta} = \mathbf{1}, \pi = 1$ (i.e. there is no filtering on which markers trace QTLs). The keyword is `BLUP`.

4.3 MCMCBLUP

Same as before, but random effects are estimated via Gibbs sampler (assuming known variances). These provides standard errors of the estimates. The keyword is `MCMCBLUP`.

4.4 PREDICT

Option PREDICT computes estimates of the prediction of phenotype given model estimates. This is useful for cross-validation, but for computation of overall individual genetic values as well, if any of $\mathbf{a}, \mathbf{d}, \mathbf{u}$ are included. Additive values would be $\mathbf{a}, \mathbf{u}$. The keyword is `PREDICT`.

For example, if you have candidates for selection, create a file with dummy phenotypes (e.g. 0) and pass them through `PREDICT`.

5 Use

5.1 Parameter file

This is an example of a typical file running a full MCMC analysis. It is quite messy `:-(`. Be careful, the order has to be kept!

```
DATAFILE
./exo_data.txt
PEDIGREE FILE
./pedigri.dat
```

```
GENOTYPE FILE
./exo_genotypes.txt
NUMBER OF LOCI (might be 0)
10946
METHOD (BLUP/MCMCBLUP/VCE/PREDICT)
VCE
SIMULATION
F
GIBBS SAMPLING PARAMETERS
NITER
10
BURNIN
2
THIN
10
CONV_CRIT (MEANINGFUL IF BLUP)
1d-4
CORRECTION (to avoid numerical problems)
1000
VARIANCE COMPONENTS SAMPLES
var2
SOLUTION FILE
solutions2
TRAIT AND WEIGHT COLUMNS
1 0 #weight
NUMBER OF EFFECTS
5
POSITION IN DATA FILE TYPE OF EFFECT  NUMBER OF LEVELS
6 cross 1
5 add_animal 2272
7 perm_diagonal 2000
8 add_SNP 0
8 dom_SNP 0
VARIANCE COMPONENTS (fixed for any BLUP, starting values for VCE)
vara
2.52d-04 2
vard
1.75d-06 2
varg
3.56 2
varp
2.15 2
vare
```

```
0.19 2
RECORD ID
5
CONTINUATION (T/F)
F
MODEL (T/F for each effect)
T T T T T
A PRIORI a
1 1
a PRIORI D
1 1
USE MIXTURE (BAYES C)
T
```

Let analyze by *logical* sections.

5.1.1 Files and input-output

This should be self-explanatory. If you do not have pedigree file, put a blank line.

```
DATAFILE
./exo.txt
PEDIGREE FILE
./pedigri.dat
GENOTYPE FILE
./exo_genotypes.txt
...
VARIANCE COMPONENTS SAMPLES
var.cage.animal.txt
SOLUTION FILE
solutions.cage.animal.txt
```

Note that the continuation file is automatically created as `parameter file_cont`.

Other files automatically created are `predictions` (if `PREDICT`) and `parameter file_EBVs` with estimated breeding values.

5.1.2 Model features

```
NUMBER OF LOCI (might be 0)
10946
METHOD (BLUP/MCMCBLUP/VCE/PREDICT)
```

```
BLUP
...
TRAIT AND WEIGHT COLUMNS
1 0 #column 0 for weight means no weight
NUMBER OF EFFECTS
5
POSITION IN DATA FILE TYPE OF EFFECT  NUMBER OF LEVELS
6 cross 1
5 add_animal 2272
7 perm_diagonal 600
8 add_SNP 0
8 dom_SNP 0
...
MODEL (T/F for each effect)
T T T T T
...
USE MIXTURE (BAYESC)
T
```

In the `TRAIT AND WEIGHT COLUMNS` the column of trait and its weight have to be specified. If the column for weight is 0, then no weight is assumed.

The number of loci is the total number of SNPs (but this is again computed from the data file).

For the methods, see above.

Write as many lines under `POSITION...` as number of effects. The `POSITION` means in which the column the effect is located in the data file (which has to be in free format, i.e., columns separated by spaces). This is irrelevant for `add_SNP` and `dom_SNP`, they are read from genotype file. The `TYPE OF EFFECT` is one of the following (with their respective keywords):

- `cross` generic cross-classified "fixed" effect
- `cov` generic covariable
- `add_SNP` additive SNP effect
- `dom_SNP` dominant SNP effect
- `add_animal` additive infinitesimal effect
- `(perm_diagonal)` generic environmental random effect

You can put in your model as many generic covariables and cross-classified "fixed" effects as you want but you can put *only one* (or none) of the other.

The `NUMBER OF LEVELS` has to be 1 for covariables (no possibility for nested covariables and the like); for the SNP effects, it is determined by the `NUMBER OF LOCI`.

The `MODEL` statement allows to quickly change the model fixing a logical variable `in_model` to true (`t`) or false (`f`). But using this feature quickly becomes confusing.

The `USE MIXTURE (BAYESC)` statement starts (if `VCE`) the BayesCPi method.

5.1.3 How to use the Bayesian Lasso

This is done adding at the end of the parameter file *exactly* the following line: `OPTION BayesianLasso Tibshirani`.

And also:

- Setting option as `VCE`
- Putting `USE MIXTURE` as F

5.1.4 MCMC and convergence features

```
GIBBS SAMPLING PARAMETERS
NITER
10000
BURNIN
2000
THIN
10
CONV_CRIT (MEANINGFUL IF BLUP)
1d-4
CORRECTION (to avoid numerical problems)
1000
```

That is, a number of iterations of 10000 with a burn-in of 2000 and a thin interval of 10. The convergence criteria `CONV_CRIT` is used for BLUP, where Gauss Seidel with Residual Update is used [7]. The `CORRECTION` is used for this same strategy. Rules of thumb are:

- For MCMC: number of iterations of 100000 and burn-in of 20000. This is a *minimum* if you include SNPs and you estimate variances. Correction every 10000 iterations.

- For BLUP (known variances): number of iterations of 10000 (it will stop before); put a convergence criteria of 10^{-8} (`1d-8`) and correction every 100 iterations. If you want a quick result, you may put a convergence criteria of 10^{-4}, this resulted in negligible errors in our work.

5.1.5 A priori and starting information

```
VARIANCE COMPONENTS (fixed for any BLUP, starting values for VCE)
vara
2.52d-04 -2
vard
1.75d-06 -2
varg
3.56 -2
varp
2.15 -2
vare
0.19 -2
RECORD ID
5
CONTINUATION (T/F)
F
...
A PRIORI a
1 10
a PRIORI D
1 1
```

Under `VARIANCE COMPONENTS` initial or a priori values are given. If the strategy is BLUP, these are the known variances; otherwise for MCMC, these are a priori distributions (inverted chi squared) for variance components. The first value is the *expectation* of the a priori distribution; the second one are the degrees of freedom. If the degrees of freedom are -2, these are “flat” (improper) distributions (roughly) equivalent to assumptions under REML.

Under `A PRIORI` the proportions of the BayesCPi mixture are given as values of the (in the example $\alpha = 1, \beta = 10$; in this order) parameters of the Beta distribution.

The `RECORD ID` is used to trace the records across the cross-validation process. This should be numeric field with a unique number for each record (not necessarily correlative).

The `CONTINUATION` statement implies this run (a MCMC one) is a continuation of a previous, interrupted one. *If this is the case*, a new file with variance components samples is created, as *variances file*`_cont`.

5.2 Pedigree file

The pedigree file has three columns: animal, sire, dam, separated by white spaces (free format). All have to be renumbered consecutively from *1* to n. Unknown parents are identified as 0. A fragment follows:

```
342     0     0
343     0     0
344     0     0
345   150   323
346   104   277
347    91   263
348    81   253
349   141   314
350   157   330
```

5.3 Data file

The format is free format (e.g. column separated by spaces). Trait values, covariables, cross-classified effects (coded from 1 to the number of levels), and the record ID can be in any order.

```
 20.3 1.08004 0.952123 1.45443 345 1 69
26.7 0.99726 1.01302 1.13901 346 2 27
19.5 1.08285 0.900454 1.33243 347 2 43
22.2 1.02697 1.01719 0.92849 348 2 2
17.3 1.05095 0.958695 1.42519 349 1 218
18.1 1.0204 1.05445 0.384847 350 2 17
25.6 0.95566 0.947974 2.06488 351 2 57
20.6 1.01382 0.921759 1.59988 352 2 36
17.3 1.01025 0.99182 1.11917 353 1 550
16.3 1.00517 0.993156 0.815969 354 2 66
```

The first four columns are the trait values, the 5^{th} column is the animal ID (coded as in the pedigree file), the 6^{th} is a cross-classified sex effect, the 7^{th} column is the "cage" effect.

5.4 Genotype file

This has to be in *fixed* format, i.e. id from column i to j and SNPs from column k to l. The format is detected by reading the first line and looking for the first space from column 50 backwards. The SNP effects have to be in one single column, coded as 0/1/2 for AA/Aa/aa (i.e., no letters, no triallelic SNP); a value of 5 implies a missing value (see below). No space is allowed among SNPs. An example (41 SNP loci) follows:

```
  45 11112121112121121102111121110101112021000
 346 11211112112110211111211121110112012021000
 347 20222222220202022020222222220202002022000
1358 11112121112121121102111121110101112021000
```

NOTE If the number of SNPs is small, the position of the last SNP will be before column 50. If this is the case, insert a fixed number of spaces, so that the position of the last SNP will be *after* column 50 and the space between ID and SNPs *before* column 50. For instance:

```
  45                                         00
 346                                         00
 347                                         20
1348                                         10
```

or

```
                                       45 00
                                      346 00
                                      347 20
                                     1348 10
```

Note that if your SNP column is buggy (less or more SNP than expected) you might have unpredictable results.

5.5 Missing values of traits or genotypes

For estimation, missing values of traits in are not allowed! Please clean your data set first. For prediction (keyword `PREDICT`), put whatever numeric column you like or a column with 0's.

If there are missing values for SNP effects, animals are set to the average of the population for additive SNP effects. Nothing is done for dominant effects (i.e., covariate is set to 0).

5.6 Variations

5.6.1 Changing random seeds

If you want to check your results with a different run, you can change the random seeds in `MODULE Ecuyer_random`, calling subroutine `init_seeds` at the beginning of the main program.

5.7 Compiling

The Fortran code is pretty standard, although some of the libraries might require some compiler switchs for portability. The main program uses a list structure using "allocatable components", aka TR 15581, which is standard in Fortran95 and available in most compilers, in particular in the free (GNU GPL licensed) compilers gfortran (>= 4.3) and g95.

5.8 Run

Running is as simple as calling it from the command line and answering about the parameter file:

```
legarra@cluster:~/mice/gsiod/gs_sparse$ ./gs3
 what parameter file?
together.cage.par
```

5.9 Output

The program does some internal checking and informative printouts, as follows:

```
 ----------------------
--        GS3       --
----------------------
     by A.Legarra
     A. Ricard, O. Filangi
     INRA, FRANCE
      03/12/2010
----------------------
03/12/2010 16:11:29
parameter file:
together.031210.par


data file:
./exo_data.txt


with:         1884 records
reading positions           6           5           7           0           0
the record id is in column           5
trait read in           1 with weight in col           0
pedigree file:
./pedigri.dat


with:         2272 records read
genotype file:
./exo_genotypes.txt
```

```
with:        1884 records read
model with          5  effects=
  -> generic cross-classified 'fixed' effect in position          6
     with          2 levels
  -> additive infinitesimal effect in position          5
     with       2272 levels
  -> generic environmental random effect in position          7
     with       2000 levels
  -> additive SNP effect in position          0
     with      10946 levels
  -> dominant SNP effect in position          0
     with      10946 levels
for a total of       26166  equations
length(in_data)=          7
reading format(i10,1x,10946i1)
--------------------------
```

With the BLUP option convergence is shown:

```
eps:    6.13867049738422
         10 ef 1 to 3   18.1022540273806        22.4239450726179
 0.764741819531106      vara,vard,varg,varp,vare,pa(1),pd(1)
 2.520000000000000E-004  1.750000000000000E-006   3.56000000000000
  2.15000000000000       0.190000000000000       0.500000000000000
 0.500000000000000
03/12/2009 08:07:07
eps:   0.953530105950441
         20 ef 1 to 3   18.1146884454257        22.4040588447695
 0.651695870345913      vara,vard,varg,varp,vare,pa(1),pd(1)
 2.520000000000000E-004  1.750000000000000E-006   3.56000000000000
  2.15000000000000       0.190000000000000       0.500000000000000
 0.500000000000000
03/12/2009 08:07:09
...
03/12/2009 08:11:48
       1382 eps  9.952282839310986E-005
solutions stored in file:
solutions.cage.animal.txt

transforming X -> divide, weighted = F
transforming yZW ->divideweighted = F
EBV's written in together.cage.par_EBVs
```

and the PREDICT option:

```
--predicting--
predicting ./exo2.txt from solutions in solutions.cage.animal.txt
 to file 'predictions'
...
predictions written
EBV's written in together.cage.predict_EBVs
--prediction finished, end of program!--
```

whereas with the MCMC option there are prints to the screen every *thin* iterations, with current samples for variance components , and the first three effects. It is interesting to check it because very high or low variances usually mean convergence problems. An example of typical output is:

```
         10 ef 1 to 3   18.1218315671272        22.4329129824538
  4.11723314223575      vara,vard,varg,varp,vare,pa(1),pd(1),includeda
 9.322796136633381E-005  2.495193547212199E-006   5.94763640217896
```

5.9.1 Solution file

The solution file name has been written in the parameter file. It looks as follows:

```
effect level  solution sderror p tau2 sdtau2
  2        1 -0.41E-02  0.24282092E-01   1.0 0.64E-03  0.66E-03
  2        2  0.40E-02  0.26491797E-01   1.0 0.71E-03  0.75E-03
   ...
```

where the effect, level and solution are self-explanatory; as for the sderror, it contains the standard error as computed by VCE or MCMCBLUP options; p is the posterior probability that the SNP is retained in the BayesC model; tau2 are the individual variances τ^2 for each SNP, as computed from Bayesian Lasso.

5.9.2 Variance components samples

Variance components, π's from BayesCPi and λ^2 from Bayesian Lasso are stored in the appropriate file, which looks as follows:

```
 vara vard varg varp vare pa_1 pd_1 2varapqpi lambda2
 0.28955E-03   0.175E-05  3.56 2.15  4.4927  1.0  1.0 1.0951 6907.3
 0.30484E-03   0.175E-05  3.56 2.15  4.2219  1.0  1.0 1.1529 6560.8
```

where we found the variance components and `pa_1`,`pd_1` are the π proportions of the mixture for non-null additive and dominant marker locus effects, respectively. Also, `2varapqpi` is actually

$$2Var(a)\pi \sum p_i q_i$$

that is, an estimator of the total genetic variance in the population [3]. This estimator is correctly computed for all cases (GBLUP with VCE, BayesCPi, Bayesian Lasso). Actually, in the Bayesian Lasso, $Var(a) = 2/\lambda^2$. You should run Post-Gibbs analysis to verify convergence using this file.

5.9.3 EBV file

A file with EBV's is always generated, with name `parameter file_EBVs`. This file contains the sum of marker locus effects for each record (identified by its id) in the data set, as well as the polygenic breeding value for that animal.

```
 id EBV_aSNP EBV_dSNP EBV_anim EBV_overall
      345  -0.593444       0.195513E-01    1.58850        1.01461
      346    1.02768       0.133699E-01    1.54519        2.58624
      347  -0.463641       0.110049E-01   -1.37548       -1.82812
      348   0.709268       0.167737E-01   -1.02831      -0.302271
      349   0.536807       0.111886E-01  -0.214559       0.333436
      350   0.343763       0.104102E-01   -3.43426       -3.08008
```

5.9.4 Prediction file

When the `PREDICT` option is requested, a file `predictions` with predictions is written; this file looks as follows:

```
id true prediction
        345  0.000000000000000E+000   20.1683639909704
        346  0.000000000000000E+000   26.5835060932076
        347  0.000000000000000E+000   19.6251279892269
        348  0.000000000000000E+000   22.1100022521052
        349  0.000000000000000E+000   17.1784939889099
        350  0.000000000000000E+000   18.2351226649716
        351  0.000000000000000E+000   25.4024678477097
```

References

[1] Gustavo de los Campos, Hugo Naya, Daniel Gianola, José Crossa, Andrés Legarra, Eduardo Manfredi, Kent Weigel, and José Miguel Cotes. Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics*, 182(1):375–385, May 2009.

[2] Rohan L Fernando. Bayesian methods in genoma association studies. Technical report, Iowa State University, 2010.

[3] Daniel Gianola, Gustavo de los Campos, William G Hill, Eduardo Manfredi, and Rohan Fernando. Additive genetic variability and the bayesian alphabet. *Genetics*, 183(1):347–363, Sep 2009.

[4] Mike Goddard. Genomic selection: prediction of accuracy and maximisation of long term response. *Genetica*, 136(2):245–257, Jun 2009.

[5] K. Kizilkaya, R. L. Fernando, and D. J. Garrick. Genomic prediction of simulated multibreed and purebred performance using observed fifty thousand single nucleotide polymorphism genotypes. *J Anim Sci*, 88(2):544–551, Feb 2010.

[6] R. Lande and R. Thompson. Efficiency of marker-assisted selection in the improvement of quantitative traits. *Genetics*, 124(3):743–756, Mar 1990.

[7] A. Legarra and I. Misztal. Technical note: Computing strategies in genome-wide selection. *J Dairy Sci*, 91(1):360–366, Jan 2008.

[8] Andrés Legarra, Christèle Robert-Granié, Pascal Croiseau, François Guillaume, and Sébastien Fritz. Improved lasso for genomic selection. *Genet Res (Camb)*, 93(1):77–87, Feb 2011.

[9] Andrés Legarra, Christèle Robert-Granié, Eduardo Manfredi, and Jean-Michel Elsen. Performance of genomic selection in mice. *Genetics*, 180(1):611–618, Sep 2008.

[10] M. Lynch and B. Walsh. *Genetics and analysis of quantitative traits.* Sinauer associates., 1998.

[11] T. H. E. Meuwissen, B. J. Hayes, and M. E. Goddard. Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, 157(4):1819–1829, 2001.

[12] T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[13] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[14] P. M. VanRaden. Efficient Methods to Compute Genomic Predictions. *J. Dairy Sci.*, 91(11):4414–4423, 2008.