GS3

Genomic Selection — Gibbs Sampling — Gauss Seidel

Andrés Legarra [*][†]

December 3, 2009

[*] andres.legarra [at] toulouse.inra.fr

[†] INRA, UR 631, F-31326 Auzeville, France

Contents

1 Introduction

This draft describes using and understanding a software for genome-wide genetic evaluations and validations, inspired in the theory by [6], and used for own our research in [4]. 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 me with the heterogenous variances.

The computing methods have been described in [3].

1.1 History

I wrote this program to implement genome-wide genetic evaluation (*aka* genomic selection) in mice [4], as there was nothing available around. The program uses Gibbs sampling, by means of an unconventional Gibbs sampling scheme [3]. It accepts quite general models.

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 [2, 6] 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. This implies that for 10000 loci and biallelic markers, 20000 effects have to be estimated. Hereinafter and for the sake of clarity we will refer to a_{jk} as *"marker locus effects"*.

3 Models

3.1 General model

The following kind of linear models is supported:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{Z}\mathbf{a} + \mathbf{W}\mathbf{d} + \mathbf{T}\mathbf{u} + \mathbf{S}\mathbf{c} + \mathbf{e} \tag{1}$$

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 $\mathbf{c}$.

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^2_u, you can use an *approximate* formula which is extensively discussed in [1]: $\sigma^2_a = \sigma^2_u / 2 \sum p_i q_i$ where p_i is the allelic frequency at SNP i.

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 [5]. 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, it comes up when the genotype is "12".

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:

$y_i | \cdots = N(\hat{y}_i, \sigma^2_e / \sqrt{\omega_i})$, 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 $\mathbf{R}_{i,i} = \sigma^2_e / \sqrt{\omega_i}$.

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 A priori information

Prior inverted-chi squared distributions can be postulated for variance components σ^2_a, σ^2_d, σ^2_u, σ^2_c, σ^2_e for estimation with `VCE`. These are also starting values.

4 Functionality

4.1 MCMC

A full MCMC is run with the keyword `VCE`. This samples all possible unknowns $(\mathbf{y}, \mathbf{b}, \mathbf{a}, \mathbf{d}, \mathbf{u}, \mathbf{c}, \sigma^2_a, \sigma^2_d, \sigma^2_u, \sigma^2_c, \sigma^2_e)$. Output are samples of variance components components 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. 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 [6]. Therefore it is an estimator that assumes known variances for all random effects. 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.txt
PEDIGREE FILE
./pedigri.dat
NUMBER OF LOCI (might be 0)
10946
METHOD (BLUP/MCMCBLUP/VCE/PREDICT)
BLUP
GIBBS SAMPLING PARAMETERS
NITER
10000
BURNIN
2000
THIN
10
CONV_CRIT (MEANINGFUL IF BLUP)
1d-4
CORRECTION (to avoid numerical problems)
1000
VARIANCE COMPONENTS SAMPLES
var.cage.animal.txt
SOLUTION FILE
solutions.cage.animal.txt
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
FORMAT
(7f12.0,1x,a21892)
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
```

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
...
FORMAT
(7f12.0,1x,a21892)
...
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.

The `FORMAT` statement has to contain a valid Fortran format. Fixed format is needed in order to read the SNPs in a simple way. In the example, there are 7 columns with numbers (either integer or real) or width 12, one space, and a single, long, chain of characters of width 21892 (i.e. twice the number of SNPs).

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
```

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.

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). 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.

5.1.3 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 [3]. 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.

5.1.4 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
```

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.

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 data file has certain restrictions, in that the format of the SNP information is somewhat specific. The format is fixed format as described by the `FORMAT`. Trait values, covariables, cross-classified effects (coded from 1 to the number of levels), and the record ID can be in any order.

The SNP effects have to be in one single column, coded as 1/2 (i.e., no letters, no triallelic SNP); a value of 0 implies a missing value (see below). No space is allowed among SNPs. The order is: first allele at the first locus, second allele at the first locus, first allele at the second locus, second allele at the second locus, and so on. An example (3 SNP loci) follows:

```
20.3    1.08004   0.952123    1.45443        345         1        69 121212
26.7    0.99726    1.01302    1.13901        346         2        27 121222
19.5    1.08285   0.900454    1.33243        347         2        43 221122
22.2    1.02697    1.01719    0.92849        348         2         2 121212
17.3    1.05095   0.958695    1.42519        349         1       218 221122
18.1     1.0204    1.05445   0.384847        350         2        17 121212
25.6    0.95566   0.947974    2.06488        351         2        57 121222
20.6    1.01382   0.921759    1.59988        352         2        36 121222
17.3    1.01025    0.99182    1.11917        353         1       550 221122
16.3    1.00517   0.993156   0.815969        354         2        66 221122
21.8     0.9588   0.981813    1.73226        355         2       418 121212
```

The first four columns are the trait values, the 4th column is the animal ID (coded as in the pedigree file), the 5th is a cross-classified sex effect, the 6th column is the "cage" effect, and the last, one, single (i.e., no space between different SNP/locus), column has the SNP codes. These are read based on the format described below as a "word" and from then considering `NUMBER OF LOCI`, so be careful when writing it. For this particular example, the syntax would be:

```
NUMBER OF LOCI (might be 0)
3
...
TRAIT COLUMN
1 0
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
```

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

5.4 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, these are not considered for that animal; in practice this assumes the animal as an heterozygote for that SNP.

5.5 Variations

5.5.1 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.

5.6 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.

5.7 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
```

5.8 Output

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

```
 BEWARE
 number of levels for effect           1 type cross               changed from
           1 to           2 line           2

 ----------------------
```

```
--        GS3       --
----------------------
     by A.Legarra
     INRA, FRANCE
      03/12/2009
----------------------
03/12/2009 08:06:17
parameter file:
together.cage.par


data file:
./exo.txt


with:         1884 records
reading positions           6           5           7           8           8
the record id is in column           5
trait read in           1 with weight in col           0
pedigree file:
./pedigri.dat


with:         2272 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           8
     with       10946 levels
  -> dominant SNP effect in position           8
     with       10946 levels
for a total of       26166  equations
length(in_data)=           7
reading format(7f12.0,1x,a21892)
--------------------------
method:
BLUP


variances: vara vard varg varp vare
 2.520000000000000E-004  1.750000000000000E-006   3.56000000000000
  2.15000000000000       0.190000000000000
residual is updated (corrected) every        1000 iterations
saving for continuation every        1000 iterations

--Gauss Seidel parameters--
convergence criterion:  1.000000000000000E-004
```

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:

```
180 ef 1 to 3  18.1659058264173900  22.3168175878622961  -0.9125053797566799
vara,vard,varg,varp,vare
2.5278284029726145E-05   4.4187941104713875E-05   5.0546789753597645   1.3128332133040825
  4.6393201375252893E-02
03/03/2008 15:22:27
181 ef 1 to 3  18.1723101703821399  22.3161450176608369  -2.2020347081531684
vara,vard,varg,varp,vare
 2.4921646267398043E-05   4.4550490791343369E-05   5.0269364529095544   1.2122964151728810
  4.3372956469675615E-02
03/03/2008 15:22:27
```

5.8.1 Solution file

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

```
effect level  solution sderror
  1         1   17.909976       0.0000000
  1         2   22.058364       0.0000000
  2         1  0.39519535       0.0000000
  2         2  0.77136298       0.0000000
```

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:

```
effect level  solution sderror
  1         1   18.931753      0.38530390
  1         2   23.084521      0.37154430
  2         1  0.39389808       1.6882820
  2         2  0.87480062       1.7113591
```

5.8.2 Variance components samples

These are stored in the appropriate file, which looks as follows:

```
 vara vard varg varp vare
   0.82285E-04   0.18682E-05    6.2945       0.54361       0.95970E-01
   0.76974E-04   0.20309E-05    6.0273       0.56775       0.98271E-01
   0.69268E-04   0.19860E-05    5.2333       0.68153       0.99926E-01
   0.68800E-04   0.19725E-05    5.0527       0.75375       0.10187
   0.62217E-04   0.19487E-05    4.7552        1.0035       0.10578
   0.60666E-04   0.20387E-05    4.4073        1.2026       0.10424
   0.61231E-04   0.19092E-05    4.6897        1.2870       0.97652E-01
```

You should run Post-Gibbs analysis to verify convergence using this file.

5.8.3 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
```

5.8.4 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] 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.

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

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

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

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

[6] 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.