TM

Threshold Model

Andrés Legarra [*][†]
INRA-SAGA, Toulouse, France

Luis Varona
Universidad de Zaragoza, Zaragoza, Spain

Evangelina López de Maturana
CNIO, Madrid, Spain

August 3, 2011

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

[†] We thank all the people who have used the program in its (permanent) buggy state, including Loys Bodin, Hervé Chapuis, Ingrid David, Miriam Piles, Anne Ricard, Jorge Urioste, Zulma Vitezica.

Contents

1 Introduction

This document describes the use and methods of a Fortran90 software for multiple trait estimation of variance components, breeding values and fixed effects in threshold, linear and censored linear models in animal breeding. The program is self-contained and quite standard so it should compile almost everywhere. It has been tested with AIX xlf90, DVF, g95 and NAG f95 for Linux.

1.1 History

The core of the program is a small multiple trait program by Luis Varona which we converted into a multiple trait 1-threshold trait program. Program grew for our own research [14, 3, 1, 5]. The software has been used in, at least, the following publications: [14, 26, 1, 16, 24, 25, 3, 5, 17, 7, 4, 18, 19, 2, 12, 9, 23]. I (AL) added many things, including multiple threshold traits, different models per trait, proper handling of conditional inverted Wishart, permanent environment, censored traits, generalized inverses, and so on. Evangelina López de Maturana added several pieces of code here and there, including the sire models and covariates. There are several subroutines taken from Ignacy Misztal's BLUPF90 distribution at `http://nce.ads.uga.edu/~ignacy`, and others are from the Alan Miller web page at `http://users.bigpond.net.au/amiller/` .

2 Functionality

The program computes:

- Posterior distributions for variance components and relevant ratios (heritabilities, correlations).
- Posterior distributions for breeding values and fixed effects with known or unknown variance components.

The program handles:

- Any number of continuous traits.
- Several continuous traits, several polychotomous traits and one binary trait.
- Several binary traits (with some restrictions).

- Theoretically, it can handle several continuous and binary traits at the same time but this can give some problems.
- Censored continuous traits.
- Missing values.
- Sire and animal models.
- User-defined covariance matrices `g_usr`
- Simultaneous correlated animal effects (e.g., sire-dam models for fertility [1] –but not “reduced animal models”– or maternal effects).
- Several random environmental effects (permanent effect).
- Different design matrices.
- It is possible to test contrasts of fixed or random effects.
- Cross-classified (i.e., effects with “levels”) and covariates.
- Optional different random seeds

The program does *not* handle:

- Nested covariates (neither random regression)
- Heterogeneous variances

2.1 Other software

There are other software doing similar tasks but not many. None (up to my knowledge) can include normal right-censored traits, except for survival traits (i.e., the Survival Kit). Van Tassell et al. MTGSAMTHR [21] can also run threshold models, but is much slower and we had numerical problems for multiple traits. Sampling of the residual covariance matrix is an approximation of unknown quality [22]. Same for GIBBS90THR1 [15], which on the other hand has a more flexible modeling (covariates, etc.). The exact method by Korsgaard et al. [10] for sampling the residual covariance is implemented here. I’ve tried to put better output (results of variance component estimates and random effects) as well as the BLUP option. Program is quite fast.

3 Methods

3.1 Gibbs sampling

MCMC and Gibbs sampling methods are used. A good reference for Gibbs sampling is Sorensen and Gianola [20]. The advantage of the MCMC and Gibbs sampling is that you can keep the same core (a “standard” multiple-trait Gibbs sampling) if you manage to integrate the liability by “data augmentation” (see the book of Tanner “Tools for statistical inference” for details). Most of the relevant theory is there. Flat priors are used for fixed effects and variance components, so the univariate estimators are equivalent to REML and the multivariate estimators are the VEIL estimators of Gianola and Foulley [6].

3.2 Threshold models

The threshold (or probit) models are quite known and well described in Sorensen and Gianola [20]. They always consider one or several thresholds and a liability that, over a given threshold, produces an observed phenotype. The key idea of the Gibbs sampler for threshold models is to include this liability as a nuisance parameter and to integrate it out in the Gibbs sampler. At each iteration, for each polychotomous record (say 0 or 1), a liability is “generated” below or over the threshold such that the observed value is 0 or 1. To avoid over/underflows, the liability is bounded between -999 and +999.

For dichotomous traits, for the parameters to be identifiable, a restriction is set so that residual variance is set to 1 and threshold is set to 0. This poses problems for multiple binary traits. For polychotomous traits, a single restriction is enough, namely, the difference between the first and second threshold is set to 1. This is more convenient computationally.

3.2.1 Restrictions in the residual covariance matrix

For binary traits, the residual variance is set to 1. Therefore, each sample of the matrix of residual covariances $\mathbf{R}_0$ has the following shape:

$$\mathbf{R}_0 = \begin{pmatrix} \sigma^2_{e11} & \sigma_{e12} & \cdots & \sigma_{e1n} \\ \sigma_{e21} & \ddots & & \\ \cdots & & & \\ \sigma_{en1} & \cdots & \cdots & 1 \end{pmatrix}$$

Therefore it is not any longer a standard inverted Wishart distribution, but a conditional inverted Wishart distribution. This is sampled according to Inge Riis Korsgaard et al. [10]. The problem is that, when there are several binary traits, the algorithm assumes that they are uncorrelated at the residual level. If there are 4 traits, the last two binary, the residual covariance matrix is forced to be:

$$\mathbf{R}_0 = \begin{pmatrix} \sigma^2_{e11} & \sigma_{e12} & \sigma_{e13} & \sigma_{e14} \\ \sigma_{e21} & \sigma_{e22} & \sigma_{e23} & \sigma_{e24} \\ \sigma_{e31} & \sigma_{e32} & 1 & 0 \\ \sigma_{e41} & \sigma_{e42} & 0 & 1 \end{pmatrix}$$

which is very unnatural. This gives also numerical problems. There are tricks to avoid this problem, but they have to be checked. One is to add an artificial environmental variance for each record, which will substitute part of the residual one (e.g., an "equivalent" model [8]). This should work for several binary traits. The other is to let the residual variance free (non identifiable). It usually does not go out of bounds and correlations and heritabilities are still identifiable. However, breeding values and fixed effects are not and (if desired) should be rescaled in each iteration. To run this trick, you need to "cheat" the program telling him that the binary trait is a 3-categories trait. The idea is by Romdhane Rekaya and can be found in J. Anim. Sci. Vol. 81, Suppl. 1: 113.

3.3 Censored traits

Censored traits are handled by "integrating" out the conditional distribution of the censored data. That means that, if we have observed a censored phenotype y^* (say interval between calvings), and we know the effects affecting this phenotype (say herd and cow), the real, unobserved phenotype y^r follows a truncated normal distribution

$$f(y^r|y^*, herd, cow) \sim N(cow + herd, \sigma^2)$$

bounded at y^*, which means that the real phenotype y^r can not be less than the observed one y^*. At each iteration of the Gibbs sampler, y^r are generated according to the values of the effects and the variances. To avoid over/underflows, y^r is also bounded between -999 and +999. The procedure is also described by Korsgaard [10]. This is "right" censoring (observed values are less than real ones). "Left" censoring is not included.

3.4 Breeding values

Breeding values are estimated. The output provides mean and standard error for all traits. They are always estimable because one genetic group is set to 0.

3.5 Fixed effects

The output also provides values for fixed effects (mean and standard errors). For cross-classified effects (e.g., season), solutions are not estimable unless the model is full rank, therefore, they should not be considered in themselves. To test fixed effects, the best is to sample them and to get contrasts (which are estimable) and their posterior distribution from these samples. It will be shown in 4.6.10 how to do that.

4 Use

4.1 Size of the problem

Implementation is with allocatable matrices and a dynamic linked list structure. Therefore there is no need to recompile the software for new problems. In principle, there are no limits for the size of the problem; but the linked list structure is slow for big problems (say 100,000 unknowns).

4.2 Pedigree file

A pedigree file *has* to be included. The pedigree file has to be numerically sorted (a typical `sort -n pedigree -o pedigree` in Unix/Linux is enough).

For animal models, the pedigree is composed of three columns, animal, sire, dam, in free format (separated by spaces). For unknown ancestors, genetic groups *must* be used. It is possible to fit only one genetic group for all unknown parents and the model is equivalent to a model without genetic groups.

For sire models, the pedigree file is of the form

```
sire, sire of sire, maternal grandsire of sire.
```

No genetic groups are allowed in this case. For unknown parents, a zero has to be used.

4.2.1 Renumbering

Animals ID have to be recodified to integer numbers ranging from 1 to the number of animals (say *nanim*). The genetic groups must be codified as *nanim+1, nanim+2*, etc. The order of the animals in the recodification does not matter.

4.3 User supplied $\mathbf{G}^{-1}$

It is possible to supply the program with a user defined *inverse* of a covariance matrix $\mathbf{G}^{-1}$ (for instance, a "genomic relationship matrix"), when the model is of type `g_usr`. You need to create and invert the covariance matrix by yourself. This file will substitute the pedigree file and has to be written as

```
i, j, value
```

where i and j are the positions in the matrix and `value` is the corresponding value. The file has to be upper or lower stored *but not full stored.* Only non-zero elements need to be stored. For instance, the inverse of the numerator relationship matrix of this pedigree:

```
1 0 0
2 0 0
3 1 2
4 1 2
```

can be stored and read as:

```
    1       1      2.0000
    1       2      1.0000
    1       3     -1.0000
    1       4     -1.0000
    2       2      2.0000
    2       4     -1.0000
    2       3     -1.0000
    3       3      2.0000
    4       4      2.0000
```

4.4 Data file

The order in the data file is important. The data file has to be arranged in columns separated by spaces, and in the following order:

```
 covariables (if any),
 fixed cross-classified effects,
 random environmental effects,
 animal genetic effects,
 continuous traits,
 polychotomous traits,
 binary traits
```

The only mandatory columns in the data file are the animal genetic effect and at least one trait.

This is an example of a data file with 4 cross-classified fixed effects, an animal effect, a continous and a binary trait.

```
legarra@cluster:~/TM$ head datoskk
  4         18       722         6      1101    462.5423       2.0000
  3         17        81         4      1102    290.9461       1.0000
  1         20       606         3      1103    344.5742       2.0000
  2          3        31         7      1104    363.5641       1.0000
 10         14       420         2      1105    400.2891       2.0000
  7         12        54         8      1106    337.3424       1.0000
  2          2       537         1      1107    387.7675       2.0000
  5         19       345         4      1108    443.7464       1.0000
  9         19        80         7      1109    367.4686       1.0000
  7         16       678         5      1110    482.7182       1.0000
```

This is an example of a data file with 1 covariable and 3 cross-classified fixed effects, an animal effect, a continous and a binary trait.

```
legarra@cluster:~/TM$ head datoskk
  0.4       18       722         6      1101    462.5423       2.0000
  3         17        81         4      1102    290.9461       1.0000
  11        20       606         3      1103    344.5742       2.0000
  0.256      3        31         7      1104    363.5641       1.0000
 -10        14       420         2      1105    400.2891       2.0000
  -7        12        54         8      1106    337.3424       1.0000
  2          2       537         1      1107    387.7675       2.0000
  0.5       19       345         4      1108    443.7464       1.0000
  0.9       19        80         7      1109    367.4686       1.0000
  70.3      16       678         5      1110    482.7182       1.0000
```

4.4.1 Renumbering

All effects (except the covariables) have to be renumbered from 1 to the number of levels.

4.4.2 Codifying of binary and polychotomous traits

Binary traits have to be codified as 1 or 2. For example 1=non pregnancy, 2=pregnancy. The 0 value is reserved to missing values. Polychotomus traits have to be codified as 1, 2, 3 .... For example, calving ease is codified as 1=no assistance, 2=slight assistance, 3=difficult, 4=very difficult.

4.4.3 Codifying of censored traits

A censored value is observed as a lower bound for the real value. For example, a cow was not pregnant 105 days after previous calving and then was sold. The lower value for days open is 105, but the real value will be higher than that (as explained previously, section 3.3). To inform the software about it, censored recordings are codified as *negative* numbers: -105 in this case.

4.4.4 Missing values

Missing values are codified as 0 (actually, any number between -0.01 and 0.01) and included in the analysis by "data augmentation" also. If you have non random missingness (a trait is observed if the other is not observed, say litter size and fertility) then the data augmentation theory does not hold and results will not make sense.

Note that there is no handling of missing covariates!

4.5 Parameter file

The program is driven by a parameter file with titles and comments. These are skipped by its position, therefore one has to be very careful when writing it. This is an example of the parameter file with 5 fixed effects. Note that the number of levels for the covariate is 1 (mandatory).

```
Data file
datoskk
Pedigree file
geneakk
Model
animal
6 Number of effects (including animal)
1 Number of covariates
1 Number of genetic groups
2 Number of traits
1 Number of threshold traits
2  Categories for the threshold traits
```

```
0 Number of random environmental effects
1 Number of animal effects
1 10 20 100 20 1000 Levels for each effect (do not include genetic group)
1  1 1  1  1     1 Model for trait 1
0  1 1  1  1     1 Model for trait 2 ... repeat as many lines as traits
Task
VCE
Total number of iterations
100000
Burn-in (discarded only in the results and solutions file)
30000
Thin interval (samples are taken every...)
100
Genetic variance
1 0
0 1
Permanent (keep always this title)
Residual
1 0
0 1
```

4.6 Variations

4.6.1 Number of iterations and burn-in

The number of iterations has to be set a priori, but one must not wait forever. Prudent guesses are (to my experience):

1. For continuous traits, 50000 iterations give a good guess and 100000 to 200000 are good enough.
2. For complex models (threshold models, maternal effects, etc) 300000 to 500000 can be enough.

Then I usually discard about 1/5 of the iterations, but this can be done using a post-gibbs software. For the thin interval, I change it to have 1000 or 5000 samples (more are hard to handle in the post-gibbs analysis and not much informative). The software prints in screen one sample every *thin* iterations. From this, the total running time can be calculated. If it is too much, just stop the program and change it.

4.6.2 Sire models

Write `sire` instead of `animal`; verify that your genealogy is in `sire-sire of sire-maternal grandsire` form; set the number of genetic groups to 0.

4.6.3 User defined $\mathbf{G}^{-1}$ g_usr

Write `g\_usr` instead of `animal`; verify that your `g\_usr` file with elements in $\mathbf{G}^{-1}$ is in the format described above (4.3); set the number of genetic groups to 0.

4.6.4 Several threshold traits

For example, 1 trait with 5 categories and 1 trait with 2 categories.

```
2 Number of traits
2 Number of threshold traits
5 2  Categories for the threshold traits
```

4.6.5 Permanent environment

It is possible to include as many permanent environmental effects as desired. In the same example, if the 4th effect is random:

```
1 Number of random environmental effects
...
Genetic variance
1 0
0 1
Permanent (keep always this title)
permanent 1
1000 0
0 1
Residual
1 0
0 1
```

Note that `permanent 1` and a corresponding matrix has to be added for each random environmental effect, i.e., if there are two:

```
Permanent (keep always this title)
permanent 1
1000 0
0 1
```

```
permanent 2
100 0
0 10
```

But if there is *no* permanent environment:

```
1 Number of random environmental effects
...
Genetic variance
1 0
0 1
Permanent (keep always this title)
Residual
1 0
0 1
```

4.6.6 Different models per trait

Say that 1st trait is affected by the 2nd effect only (not even the animal effect!). The program sets those effects to zero. This works for any trait/effect combination, including random and genetic effects.

```
10 20 100 10 1000 Levels for each effect
  0 1  0  0     0 Model for trait 1
  1 1  1  1     1 Model for trait 2 ...
```

4.6.7 Variance components or breeding values

We can estimate genetic parameters as shown, or we can estimate breeding values with fixed variance components (BLUP) if use the word `BLUP` instead of `VCE`. Evangelina López de Maturana uses this option to get breeding values of calving ease in dairy cattle.

4.6.8 Covariance matrices

The covariance matrices which are included at the parameter file are used as known if we are running BLUP. If not, they are used as starting points. Zeros out of the diagonal do not imply the covariance is set to zero. There are two particular cases in which they have to be well chosen:

- If we are in a sire model, where $\sigma_s^2 < \frac{1}{4}\sigma_e^2$.

- When there are censored traits, the censored value has to be "likely" under the variance chosen. That is, if we see values of 54, the variance should not be 1. Without this caution, the program gets stuck trying to sample "real" records. The best is to use the phenotypic variance or something similar.

For models with different random matrix per trait, the program handles them well because it uses generalized Inverted Wishart based on generalized inverses. For example, for this model,

```
10 20 100 10 1000 Levels for each effect
  0 1  0  0     0 Model for trait 1
  1 1  1  1     1 Model for trait 2 ...
```

the genetic variance is only defined for trait 2. The program produces the output

```
  0.0000000000000000E+000  0.0000000000000000E+000
  0.0000000000000000E+000  35.2971843629242059
```

therefore $\sigma^2_{a2,2} = 35.29$ and the rest is zero. For multiple animal effects the genetic variances as organized traits within effects. That is, for a bull-cow model for fertility and 2 traits (say, days open and success at first insemination), the genetic covariance is:

$$G_0 = \begin{pmatrix} A & B \\ B' & D \end{pmatrix} = \left(\begin{array}{cc|cc} \sigma^2_{a11} & \sigma_{a12} & \sigma_{a13} & \sigma_{a14} \\ \sigma_{a21} & \sigma^2_{a22} & \sigma_{a23} & \sigma_{a24} \\ \hline \sigma_{a31} & \sigma_{a32} & \sigma^2_{a33} & \sigma_{a34} \\ \sigma_{a41} & \sigma_{a42} & \sigma_{a43} & \sigma^2_{a44} \end{array}\right)$$

Then A is the matrix of genetic covariances of the effect bull for the traits days open and success insemination; B is the covariance between bull and cow effects for those traits; and D is the covariance matrix of the effect cow for the traits days open and success at first insemination.

4.6.9 Maternal effects or several animal effects

It is possible to include several animal effects (e.g., maternal effects or bull - cow models in fertility), for example: `2 Number of animal effects`. We will need to put them correctly in the effects part of the model. The same pedigree is assumed for all of them. Note that sire models for maternal effects model are also possible, although backtransforming the sire variances into genetic variances is quite awful [11]. We have done it with good results.

For a model with two traits and maternal effects affecting the second, this is a parameter file:

```
Data file
datoskk
Pedigree file
geneakk
Model
animal
5 Number of effects (including animal)
1 Number of genetic groups
2 Number of traits
1 Number of threshold traits
2  Categories for the threshold traits
1 Number of random environmental effects
2 Number of animal effects
100 10 100 1000 1000 Levels for each effect
  1  1 0    1    0 Model for trait 1
  1  1 1    1    1 Model for trait 2 ...
Task
VCE
Total number of iterations
100000
Burn-in (discarded only in the results and solutions file)
30000
Thin interval (samples are taken every...)
100
Genetic variance
1 0 0 0
0 1 0 0
0 0 0 0
0 0 0 1
Permanent (keep always this title)
permanent 1
0 0
0 1
Residual
1 0
0 1
```

Note that this model includes one random environmental effect (dam, non genetic), a genetic effect for both traits (individual) and a genetic effect for the second trait (dam). The order of the genetics effects does not matter. The data file is:

```
legarra@cluster:~/TM$ head datoskk
 60          7          25          111          35     388.2996         1.0000
  2          4          69          112          79     390.9525         2.0000
 74          9          86          113          96     449.0446         2.0000
 72          9          63          114          73     366.7321         1.0000
 25          8          68          115          78     453.1664         2.0000
 96         10          68          116          78     364.5786         1.0000
 70          5          35          117          45     427.0817         1.0000
  4          3          81          118          91     323.0574         1.0000
 61         10          63          119          73     318.4384         1.0000
 95          3          21          120          31     343.8603         1.0000
```

Note that the dam environmental effect (3rd column) has to be renumbered and this number is not the same as the one in the genetic effect (5th column), because there are less levels. The 4th column is the individual. One sample of the genetic covariance matrix is:

```
  37.7116  37.3271  0.00000000E+000  -6.75222
  37.3271  65.8084  0.00000000E+000  -4.90055
  0.000000E+000  0.00000E+000  0.000000E+000  0.0000E+000
  -6.75222  -4.9005  0.0000000E+000   3.37377
```

which shows that the genetic variance component of the dam for the 1st trait is zero.

4.6.10 Contrasts

It is hard to think in a standard type of contrasts, so this is the way to program them. The idea is to print out samples of the vector of solutions to a file, just as the variance components are. Look for this section:

```
! ---------
! Contrasts
! ---------
!    uncomment next line if you want contrasts
!    write(20,'(20f15.8)') b(31:33,1),b(31:33,2)
! -------------
! end contrasts
! -------------
```

This prints out to unit 20 ('samplesFE.txt') the solution vector (b) in the positions 31 to 33 for the 1st and 2nd trait. The positions are obtained by the sum of the levels of all the previous effects, plus the level we are interested in. In the example in 4.5 this corresponds to the 1st to 3rd level of the 3rd effect. Other way of doing the same is using the vector ifac which stores the starting address of each effect. For the same example, this would be:

```
write(20,'(20f15.8)') b( (ifac(3)+1):(ifac(3)+3),1), &
                      b(ifac(3)+1):(ifac(3)+3),2)
```

If you do not want this output, just comment it (as it is usually).

4.6.11 Changing random seeds

If you want to check your results with a different run, you can change the random seeds by using an OPTION statement. This is a line at the end of the program as follows:

```
OPTION RandomSeeds seed1 seed2 seed3
```

where `seed1 seeds2 seed3` are (positive or negative) integers, for instance

```
OPTION RandomSeeds 1234 -5687 -986
```

Please respect the case of `OPTION RandomSeeds` or it will not work.

4.7 Compiling

4.7.1 NAG for Linux

There is a fairly good amount of legacy code and compilers might complain about that. To compile it in the Linux cluster of INRA-SAGA (NAG compiler):

`legarra@cluster:~/TM$ f95 -O3 -o TM tm.f90` . As this is Gibbs sampling, speed matters. An optimization option (say, `-O3`) may run much faster depending on the compiler.

4.7.2 AIX xlf

Two ways to compile in using xlf90 for AIX:
`dga2:/utou/utouale/TM # f90 tm.f90 -o TM`,
or change the extension of the program to `.f` and:
`dga2:/utou/utouale/TM # xlf90 tm.f -o TM`.

4.7.3 g95 for Windows XP

This compiler is free (comes with GNU license) and available for many other operating systems. Other option is `gfortran`.

```
C:\Documents and Settings\TM>g95  tm.f90 -O3 -o TM
```

produces an executable TM.exe.

4.8 Run

Just write the name of the executable and answer:

```
legarra@cluster:~/TM$ ./TM
 -----------------
 TM - 31 October 2008
 by A Legarra, L Varona, E Lopez de Maturana
 -----------------
 started:
 date: 31/10/2008
 time: 11:30:05
 Parameter file?
simul.par
 simul.par
 number of traits with var(e) constrained to 1 --> 0
```

or “echo” it: `echo simul.par|./TM`

4.9 Output

There are prints to the screen every *thin* iterations. The print gives time and the present sample of covariance components (in the order: genetic, environmental, residual). It is interesting to check it because very high or low variances usually mean convergence problems.

```
   1.469048E+02  -0.74877  0.E+000   0.95498
  -0.74877   0.288388  0.E+000  -0.167496
  0.E+000  0.E+000  0.E+000  0.E+000
   0.95498  -0.16749  0.E+000   0.137198
  0.E+000  0.E+000
  0.E+000   8.610746E-03
   1.941E+03   1.4602
   1.4602   6.130E-02
  imue  1
```

```
 05/04/2006 14:21:23
```

4.9.1 results.txt

This is a file produced every $100 \cdot thin$ iterations after burn-in, which gives the present estimates (mean and standard errors) for variance components and genetic correlations, heritabilities, etc, after discarding burn-in. The h2, hp12, he2 stand for the ratios (and associated correlations) of total variance due to additive, permanent and residual effects. This is an extract:

```
Parameter file: simul.par
 Iteration number:     10000
 Burn-in:      3000

 Average additive variance
     7.59901448    -0.16399846     0.00000000     0.01532234
    -0.16399846     0.27350263     0.00000000    -0.15480405
     0.00000000     0.00000000     0.00000000     0.00000000
     0.01532234    -0.15480405     0.00000000     0.08896942
  Sd Additive variance
     3.22667610     0.34369945     0.00000000     0.19072825
     0.34369945     0.01706567     0.00000000     0.01195900
     0.00000000     0.00000000     0.00000000     0.00000000
     0.19072825     0.01195900     0.00000000     0.00989067

 Average environmental variance    1-th
     0.00000000     0.00000000
     0.00000000     0.02117482
  Sd environmental variance
     0.00000000     0.00000000
     0.00000000     0.00615793

  Average residual variance
  2035.84818836     0.79387308
     0.79387308     0.06443367
  Sd residual variance
   107.81988846     0.59492772
     0.59492772     0.00685906

  Average h2 and additive correlation
     0.00373137    -0.11954332     0.00000000     0.01751802
    -0.11954332     0.00000000     0.00000000    -0.99344407
     0.00000000     0.00000000     0.00000000     0.00000000
     0.01751802    -0.99344407     0.00000000     0.00000000
  Sd h2 and additive correlation
     0.00160243     0.21534016     0.00000000     0.21255278
     0.21534016     0.00000000     0.00000000     0.00502002
     0.00000000     0.00000000     0.00000000     0.00000000
     0.21255278     0.00502002     0.00000000     0.00000000
```

For sire models all this correlations are not correctly calculated and therefore not printed, because the sire variance is 1/4 of the genetic variance. They have to be inferred from the covariance samples. Same for maternal models or sire-dam models.

4.9.2 solutions.txt

This file produced every $100 \cdot thin$ iterations after burn-in contains the solutions (mean and standard error, after burn-in) for fixed and random effects, in order. The file is organized in columns, the first one is the solution for the 1st trait, the 2nd one its s.e., the 3rd is the solution of the same effect for the 2nd trait, the 4th its s.e., and so on. To get the breeding values you must start from the corresponding level, i.e., the sum of the levels of all the previous effects. In the example in 4.5 the line for the 1st breeding value is $1 + 10 + 20 + 100 + 10 + 1 = 141$. This is an example of the file:

```
548.93036050    45.16501306     1.64939092     0.27237214
537.20447345    45.45495594     1.80820386     0.32165669
556.11838284    44.45897965     1.88057097     0.29986198
545.63172656    44.45013299     1.90946556     0.26104840
542.96121989    44.92410454     1.64755362     0.39665811
550.09841527    44.20431687     1.88156240     0.30906273
```

4.9.3 thresholds.txt

This file is produced every *thin* iterations, and gives the samples of the thresholds, plus an $+\infty$ threshold which is set to 999. It is not of much interest because, for binary traits, the threshold is fixed to 0, and for polychotomous traits the first threshold is 0 and the second threshold is 1. Therefore it may be of interest for traits with more than 3 levels. The thresholds are ordered threshold within trait. Each line is one sample taken every *thin* iterations.

4.9.4 samples.txt

This is the file with samples from the posterior distribution of variance components. Each line is one sample. There is a header file indicating what is each column, e.g, this fragment is the first row of the genetic covariance matrix:

```
vara_0101       vara_0102       vara_0103       vara_0104
337.61511729    -0.65333460     0.00000000      1.49091783
273.35178896     0.63688959     0.00000000      0.30858498
384.97065327     2.17405027     0.00000000      0.58115261
342.89653329     6.01183500     0.00000000     -1.58711165
342.91126139     9.17706121     0.00000000     -1.79774013
299.79676759    -1.19988111     0.00000000      0.21024740
```

And, for example, `varp0i_jk` is the environmental covariance of the i-th random environmental effect for the traits j and k. If $j = k$, it is the variance.

4.9.5 samplesFE.txt

This is the file with samples from the posterior distribution of fixed (or random) effects if desired as explained in 4.6.10). Each line is one sample. There is no header line. It looks like:

```
  101.61328547   100.81883503   100.06813214     1.88082650     2.01280556     1.98134262
  101.10080979   100.66554871   100.35605747     1.91243514     2.01313644     1.97147973
  101.26135644   100.61985381   100.70051116     1.95634873     2.00828649     1.99967744
```

Following the example in 4.6.10, the first three columns correspond to samples of the solutions for the 3 levels of fixed effects for the first trait, and the second three columns to the solutions for the second trait. For fixed effects, these are non-estimable parameters and therefore meaningless. To do a proper analysis one needs to compute the contrasts, which are estimable functions, e.g., in SAS:

```
data one;
infile 'samplesFE.txt';
input age1 age2 age3;
contrast1=age2-age1;
contrast2=age3-age1;
run;
```

4.10 Post-gibbs analysis

Although results.txt provides a lot of information, it is important to check the Gibbs sampler and to get plots, etc. This can be done in several ways. One is to use SAS to get means and s.e. of the variance components and their functions. To compute features of functions of variance components, compute the function (say, h^2) for each sample and you get the posterior distribution of h^2. This procedure is statistically correct and much easier than using Taylor expansions. Another nice thing is that you get more precise confidence intervals and perhaps non-symmetric intervals (no more genetic correlations of 0.9 ± 0.10 beyond the bounds). To test if a correlation is different from zero one can just count how many times was it greater than zero in the posterior distribution. For example, to compute the s.e. of the heritability one can do the following in SAS:

```
data one;
infile 'samples.txt';
input vara varp vare;
* discard burn-in;
if _N_>1000;
h2=vara/(vara+vare+varp);
* get features of the posterior distribution of h2;
proc univariate plot;
var h2;
run;
```

Or to compute the posterior distribution of the contrast:

```
data one;
infile 'samplesFE.txt';
input age1 age2 age3;
* discard burn-in;
if _N_>1000;
contrast1=age2-age1;
contrast2=age3-age1;
* get features of the posterior distribution of the contrast;
proc univariate plot;
var contrast1 contrast2;
run;
```

4.10.1 R and BOA

The best is usually to use R (or S-plus) and BOA (they are in the cluster and there is R free for Windows). First some simple analysis in R:

```
> a=read.table("samples.txt",header=TRUE)
> summary(a)
   vara_0101        vara_0102        vara_0202        varp01_0101
 Min.   :0.1364   Min.   :0.1045   Min.   :0.1864   Min.   :0.05019
```

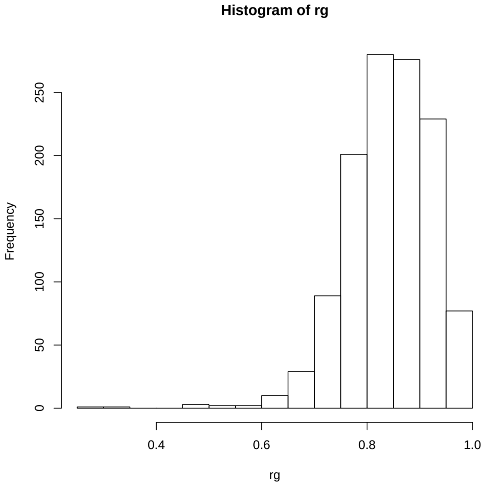

Figure 1: Histogram

```
 1st Qu.:0.2001   1st Qu.:0.1855   1st Qu.:0.2394   1st Qu.:0.23435
 Median :0.2244   Median :0.2027   Median :0.2577   Median :0.37637
 Mean   :0.2261   Mean   :0.2027   Mean   :0.2594   Mean   :0.39773
 3rd Qu.:0.2506   3rd Qu.:0.2196   3rd Qu.:0.2779   3rd Qu.:0.54658
 Max.   :0.4718   Max.   :0.3246   Max.   :0.3530   Max.   :0.80572
  ...
> # the genetic correlation
> rg=a$vara_0102/sqrt(a$vara_0101*a$vara_0202)
> summary(rg)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2683  0.7932  0.8475  0.8423  0.9018  0.9750
> hist(rg)
```

BOA (google for “Bayesian Output Analysis Program”) is a specialized package of R for MCMC output checking with many options. It is best not to include too many variables in BOA at the same time because you don’t see anything in the plots. BOA is useful for:

- Checking convergence visually and numerically.
- Plotting.

The file samples.txt has a good format for BOA. What I usually do is:

- Checking convergence by plotting running means, traces, and computing statistics (usually Heidelberg and Welch). The best is to plot correlations, which are harder to estimate.

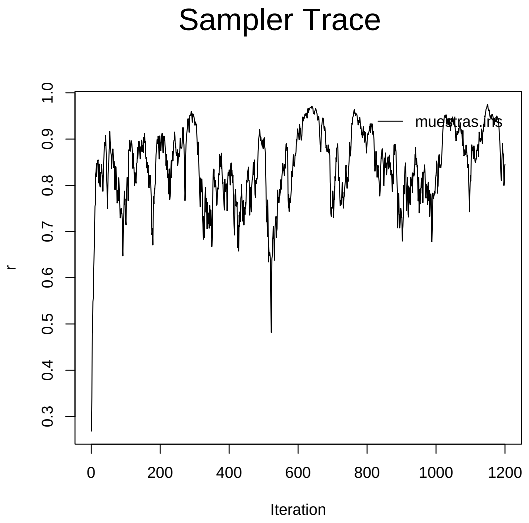

Figure 2: Trace

- Get means and relevant percentiles
- Plot graphs

After opening R, you start boa by `> boa.menu()`. Then there are menus. You read the samples.txt file by:

```
   BOA MAIN MENU -> file -> import data -> Options
     -> Working directory

Enter new character string
1: C:\Documents and Settings\legarra\Mes documents\manualTM

   BOA MAIN MENU -> file -> import data -> Flat Ascii file
Enter filename prefix without the .txt extension [Working Directory: ""]
1: "samples"
```

It is very important to enter the working directory. Then you can plot and check following the menu and BOA manual. This is an example of plots of genetic correlation between mammary insertion in first and later parities [14].

4.11 Problems

The main problems come from mistakes or very complex models. The good thing is that when Gibbs sampling does not work, it is obvious (for example, $h^2 = 0.99$). The bad thing is that problems take long to show and usually is an awful numerical error.

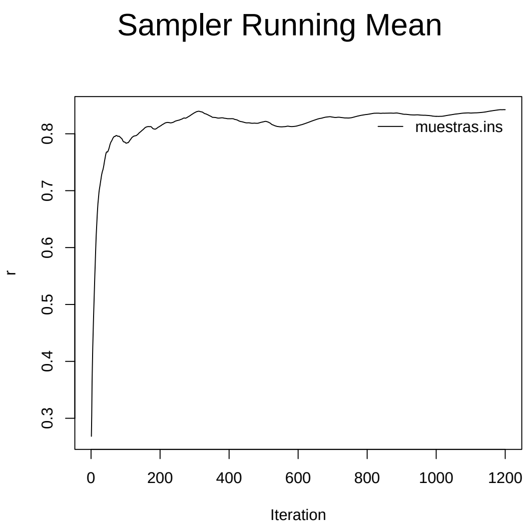

Figure 3: Running mean

Mistakes It is important to verify that codifying is correct, parameter file is good and the data and pedigree files are correct.

Cycling In very complex models (sire maternal models for several polychotomous traits) programs cycled. This can be seen by plotting traces in BOA. This was solved by using a better random number generator by L'Ecuyer.

Positive-definiteness In complex models matrix of variance components might be non-positive definite. This will give numerical problems. This might be solved by "bending" but it is not very nice because we are forcing them.

Complex models They take long time to run, are prone to errors and may not run at all. Sometimes it is better to move to other models (sire models for example).

Binary traits Binary traits may go out of bounds. The liability can be sampled very far away from the threshold if the breeding values are very high. This leads to big additive variances, which lead to big breeding values ... To avoid this, a good solution is to change to sire models. Another one is that the liability in binary traits may be set to at maximum ±4 residual standard deviations from the current mean (change it by the `liabilitybound` variable in the program). Other option is not set the residual variance to 1 (3.2.1). Multiple binary traits may produce non-positive definite matrices. Some tricks to avoid this were described in 3.2.1. Use with caution. At some point, problems come mainly from lack of good data and there is no simple solution.

Extreme case problem If there is an uneven distribution of phenotypes in one class of a fixed effect (that is, one herd with all calving ease=1), its effect is non estimable. It is recommended to fit it as a random effect. This was reported, for example, by Misztal et al. (JDS 72:1557) and Carlos Moreno (GSE 29:145).

4.12 Example and test

A simulated bivariate data set ($r_g = -0.5$) is provided for animal (data file, `datoskk` pedigree file `geneakk`) and sire `datoskksire` `geneakksire`models. The corresponding parameter files are `simul.par` and `simulsire.par`. This is a typical output from running `simul.par`:

```
legarra@cluster:~/TMdist$ echo simulsire.par ./tm
simulsire.par ./tm
legarra@cluster:~/TMdist$ echo simulsire.par | ./tm
 Parameter file?
 simulsire.par
 number of records 8900
 number of covariables:  1
 number of observed traits 2
 number of traits with var(e) constrained to 1 --> 0
 Animals, unknown parent groups = 100 0
 Model for trait 1
 1 1 1 1 0 1
 Model for trait 2
 0 1 0 1 1 1
 Estimating variance components
 Total n of iterations: 11000
 Burn-in: discarded for results.txt and solutions.txt 1000
 Thin interval 100
 Total n of samples in the Gibbs sampler: 110
 sire model
 number of records, non-null elements =  8900 29628
  11.3561111793608251  -0.3476993336463193
  -0.3476993336463193   4.3448850123729565E-02
  55.1171609582973758  -0.5592405287514247
  -0.5592405287514247   0.2152450780010031
  imue   1
 31/10/2008 10:19:21
```

with solutions file

```
     1.06141889     0.26566714     0.00000000     0.00000000
     6.32449195     2.83650750     1.42324365     0.31620909
     6.29836679     2.82475144     1.44403390     0.31708041
     5.78829387     2.82585841     1.47732260     0.31485805
     6.28683428     2.84174016     1.45555319     0.31460713
     6.52408758     2.81833951     1.47606107     0.31749095
     6.41384068     2.78175451     1.46082427     0.31747790
     6.20643893     2.87275380     1.46963281     0.31477530
     6.50733017     2.80505533     1.46167945     0.31717126
     6.70572253     2.83513320     1.44402953     0.31537910
     5.96764855     2.83464849     1.46578857     0.31286806
     4.52275743     2.91573995     0.00000000     0.00000000
     4.44778629     2.88866689     0.00000000     0.00000000
     4.96364402     2.90414365     0.00000000     0.00000000
     4.38549987     2.86524471     0.00000000     0.00000000
     4.92880843     2.94032235     0.00000000     0.00000000
```

and final result (note that the number of iterations is *too small*):

```
Parameter file: simul.par
 Iteration number:     11000
 Burn-in:      1000

 ve stat=true 0
 vp stat=true 0
 va stat=true 0
 Average additive variance
    28.11639263    -0.83182075
    -0.83182075     0.11718751
  Sd Additive variance
     1.95622311     0.09656303
     0.09656303     0.00743259

  Average residual variance
    33.72525120     0.06169531
     0.06169531     0.12649562
  Sd residual variance
```

```
     1.42452324     0.06349272
     0.06349272     0.00476781

  Average h2 and additive correlation
     0.45440480    -0.45818941
    -0.45818941     0.48062477
  Sd h2 and additive correlation
     0.02659841     0.04318270
     0.04318270     0.02398042

  Average he2 and residual cor
     0.54559520     0.03032925
     0.03032925     0.51937523
  Sd he2 and residual cor
     0.02659841     0.03138916
     0.03138916     0.02398042
```

You should obtain the same results as far as you do not change random seeds in module `MODULE Ecuyer_random`.

5 Appendix: how the linked list works

This is basically a reminder for myself. I (AL) do not know the origin of the linked list, LV already had it. A sparse matrix **B** is stored as follows:

1. `zhz`: values in **B**.
2. `iplace`:auxiliary variable.
3. `ifirst(i)`: points to the first stored element of row i. To start looping through the row, auxiliary variable is set as `iplace=ifirst(i)`.
4. `ivcol(iplace)`: which column are we at.

Thus `zhz(iplace)` stores the i,j where j =`ivcol(iplace)` element in **B**.

`inext(iplace)` indicates where is the next element in the same row. When this is 0, this is the end of the row.

So one round of Gauss Seidel for $\mathbf{B}\ \mathbf{a} = \mathbf{xy}$ is:

```
do i=1,neq
  rhs=xy(i)
  iplace=ifirst(i)
  do
    ! correcting
    rhs=rhs-zhz(iplace)*sol(ivcol(iplace))
    ! catch diagonal element
    if(i==ivcol(iplace)) lhs=zhz(iplace)
    ! go to next element
    iplace=inext(iplace)
    ! test for end
    if(iplace==0) exit
  enddo
  rhs=rhs+lhs*sol(i)
  sol(i)=rhs/lhs
enddo
```

Interestingly, converting this to `ia ja a` format should be fairly simple.

6 References

References

[1] I. David, L. Bodin, G. Lagriffoul, C. Leymarie, E. Manfredi, and C. Robert-Grani. Genetic analysis of male and female fertility after

artificial insemination in sheep: Comparison of single-trait and joint models. *J Dairy Sci*, 90(8):3917–3923, Aug 2007.

[2] I. David, MJ Carabaño, L. Tusell, C. Diaz, O. González-Recio, E. Lopez de Maturana, M. Piles, E. Ugarte, and L. Bodin. Product versus additive model for studying artificial insemination results in several livestock populations. *Journal of Animal Science*, 89(2):321, 2011.

[3] E. López de Maturana, A. Legarra, L. Varona, and E. Ugarte. Analysis of fertility and dystocia in holsteins using recursive models to handle censored and categorical data. *J Dairy Sci*, 90(4):2012–2024, Apr 2007.

[4] EL De Maturana, D. Gianola, GJM Rosa, and KA Weigel. Predictive ability of models for calving difficulty in us holsteins. *Journal of Animal Breeding and Genetics*, 126(3):179–188, 2009.

[5] H. Garreau, SJ Eady, J. Hurtaud, and A. Legarra. GENETIC PARAMETERS OF PRODUCTION TRAITS AND RESISTANCE TO DIGESTIVE DISORDERS IN A COMMERCIAL RABBIT POPULATION. In *Proceedings, 9th World Rabbit Congress 2008, June 10-13, 2008, Verona , Italy, http://world-rabbit-science.com/.*, 2008.

[6] D. Gianola and J. L. Foulley. Variance estimation from integrated likelihoods (veil). *Gen Sel Evol*, 22:403:418, 1990.

[7] O. González-Recio, E. López de Maturana, and JP Gutiérrez. Inbreeding depression on female fertility and calving ease in spanish dairy cattle. *Journal of dairy science*, 90(12):5744–5752, 2007.

[8] C. R. Henderson. Equivalent linear models to reduce computations. *J Dairy Sci*, 68:2267:2277, 1985.

[9] S. Karoui, C. Díaz, M. Serrano, R. Cue, I. Celorrio, and M.J. Carabaño. Time trends, environmental factors and genetic basis of semen traits collected in holstein bulls under commercial conditions. *Animal Reproduction Science*, 2011.

[10] Inge Riis Korsgaard, Mogens Sand Lund, Daniel Sorensen, Daniel Gianola, Per Madsen, and Just Jensen. Multivariate bayesian analysis of gaussian, right censored gaussian, ordered categorical and binary traits using gibbs sampling. *Genet Sel Evol*, 35(2):159–183, 2003.

[11] L. A. Kriese, J. K. Bertrand, and L. L. Benyshek. Age adjustment factors, heritabilities and genetic correlations for scrotal circumference

and related growth traits in hereford and brangus bulls. *J Anim Sci*, 69(2):478–489, Feb 1991.

[12] R. Lavara, JS Vicente, and M. Baselga. Genetic parameter estimates for semen production traits and growth rate of a paternal rabbit line. *Journal of Animal Breeding and Genetics*.

[13] A. Legarra, I. Misztal, and J. K. Bertrand. Constructing covariance functions for random regression models for growth in gelbvieh beef cattle. *J Anim Sci*, 82(6):1564–1571, Jun 2004.

[14] A. Legarra and E. Ugarte. Genetic parameters of udder traits, somatic cell score, and milk yield in latxa sheep. *J Dairy Sci*, 88(6):2238–2245, Jun 2005.

[15] I. Misztal, S. Tsuruta, T. Strabel, B. Auvray, T. Druet, and D.H. Lee. Blupf90 and related programs (bgf90). In *7th World Congress on Genetics Applied to Livestock Production*, pages CD–ROM Communication N 28–07, 2002.

[16] Sandrine Paget, Zulma G Vitezica, Franois Malecaze, and Patrick Calvas. Heritability of refractive value and ocular biometrics. *Exp Eye Res*, Nov 2007.

[17] A. Ricard and A. Legarra. Validation of models for analysis of ranks in horse breeding evaluation. *Genetics Selection Evolution*, 42(1):1–10, 2010.

[18] V. Riggio, R. Finocchiaro, and SC Bishop. Genetic parameters for early lamb survival and growth in scottish blackface sheep. *Journal of animal science*, 86(8):1758, 2008.

[19] V. Riggio, B. Portolano, H. Bovenhuis, and S.C. Bishop. Genetic parameters for somatic cell score according to udder infection status in valle del belice dairy sheep and impact of imperfect diagnosis of infection. *Genetics Selection Evolution*, 42(1):1–9, 2010.

[20] Daniel Sorensen and Daniel Gianola. *Likelihood, bayesian and MCMC methods in quantitative genetics*. Springer, 2002.

[21] C. P. Van Tassell and L. D. Van Vleck. Multiple-trait gibbs sampler for animal models: flexible programs for bayesian and likelihood-based (co)variance component inference. *J Anim Sci*, 74(11):2586–2597, Nov 1996.

[22] C. P. Van Tassell, L. D. Van Vleck, and K. E. Gregory. Bayesian analysis of twinning and ovulation rates using a multiple-trait threshold model and gibbs sampling. *J Anim Sci*, 76(8):2048–2061, Aug 1998.

[23] L. Tusell, A. Legarra, M. García-Tomás, O. Rafel, J. Ramon, and M. Piles. Different ways to model biological relationships between fertility and ph of the semen in rabbits. *Journal of animal science*, 89(5):1294, 2011.

[24] J. I. Urioste, I. Misztal, and J. K. Bertrand. Fertility traits in spring-calving aberdeen angus cattle. 1. model development and genetic parameters. *J Anim Sci*, 85(11):2854–2860, Nov 2007.

[25] J. I. Urioste, I. Misztal, and J. K. Bertrand. Fertility traits in spring-calving aberdeen angus cattle. 2. model comparison. *J Anim Sci*, 85(11):2861–2865, Nov 2007.

[26] Z. G. Vitezica, C. R. Moreno, L. Bodin, D. Franois, F. Barillet, J. C. Brunel, and J. M. Elsen. No associations between prp genotypes and reproduction traits in inra 401 sheep. *J Anim Sci*, 84(6):1317–1322, Jun 2006.