1 Tutorial of use of the blupf90 suite for (SSG)BLUP of line crosses (with metafounders)

Andres Legarra[1] started 24/04/2020

- version 0.1 May 7th 2020
- version 0.2 Jul 27th 2020, including some suggestions by Daniela Lourenço

1.1 Purpose

This is a tutorial on the use of the `blupf90` family of software to analyze crossbred data in SSGBLUP using the metafounders theory. First, we fit two models without markers (BLUP and BLUP with UPGs). Then we proceed to include markers and metafounders in a SSGBLUP. We use two specific binaries that are available on the blupf90 page: `renumf90` and the new `blupf90test` which also supports REML. We also use the (not yet publicly released) `gammaf90` software to compute the gamma matrix of relationships across metafounders; however, this is not needed because gamma can be computed using own software. Other binaries that are mainly used to estimate variance components like `gibbs*f90` or `*remlf90` do not consider metafounders, but in most cases `blupf90test` can be used instead.

This project has received funding from the European Unions' Horizon 2020 Research & Innovation programme under grant agreement N°772787 -SMARTER.

The files to follow this tutorial are in http://genoweb.toulouse.inra.fr/~alegarra/ThreeWayDist

1.2 Simulation

I simulated a 3-way cross using QMsim (Sargolzaei and Schenkel 2009):

```
      A x B
        |
   C x F1
     |
   3-way
```

The trait has $h^2 = 0.5$. Lines (breeds) A and C are selected for high values, line B for low values. All lines are related because they come from the same ancestral population 15 generations ago - however, we only have the last 5 generations of pedigree and data. Lines A, B and C are in one farm (with simulated farm effect +20). F1 and 3-way are in another two farms (with simulated farm effects +30 and +40 respectively). Genetic correlation across farms and populations is 1 - but variances are different. Crude averages of genetic value (gv) and phenotypic value (y) of the lines at the 1st 1000 records are:

Table 1: means.

line	$\bar{y}$	$\overline{gv}$
line A	27.54	7.54
line B	13.54	-6.46
line C	28.55	8.55
F1	30.34	0.34
3-way	46.37	6.37

The genetic scenario is probably not much realistic (*and this simulated data should not be used to prove or disprove any model*) but it allows to show the details and possibilities in a 2- and

[1] andres.legarra at inrae.fr

3- way genetic evaluation where lines are more (A and C) or less (B) similar and individuals of different breeds have different environmental effects. We simulated a single, additive trait with no heterosis. However, GxG and GxE can be accommodated in this case fitting one trait per population (i.e. 5-trait analysis). Heterosis can be modelled fitting homozygosity (inbreeding) as a covariate.

1.2.1 Pedigree

This is how the pedigree file `ped` looks like:

```
A40225 0 0 A
A40922 0 0 A
A37199 0 0 A
...
A61165 A55264 A57020 A
A61166 A55264 A57020 A
...
B101016 0 0 B
B97504 0 0 B
...
B121166 B115792 B114702 B
B121167 B115792 B114702 B
...
C160714 0 0 C
C157689 0 0 C
...
C181173 C176101 C174591 C
C181174 C176101 C174591 C
...
AxB181205 A55397 B117086 AxB
AxB181206 A55397 B117086 AxB
...
CxF1189225 C176649 AxB182251 CxF1
CxF1189226 C176649 AxB182251 CxF1
```

The pedigree has 89260 animals and it is complete in the sense that all sires and dams figure in the 1st column, with unknown (0) parents if necessary. We have a single file for pedigree including pure lines A, B and C and F1 (F1 is the cross of A x B) and 3-way (male C and female F1) crosses, which have ancestors of the different lines. The 4th column indicates the breed of the animal. The pedigree (all animals confounded) has ~5 generations and maximum inbreeding in the pure lines of 7%.

1.2.2 Data

This is how the data file `data` (29333 records) looks like:

```
A41175 M 27.5509 A PurebredFarm
A41178 F 28.4484 A PurebredFarm
...
B101241 F 14.348 B PurebredFarm
B101244 M 13.8164 B PurebredFarm
...
C161268 M 28.845 C PurebredFarm
C161271 M 30.8888 C PurebredFarm
...
AxB181308 F 30.86 AxB F1Farm
AxB181311 M 30.7664 AxB F1Farm
```

```
...
CxF1189305 M 35.366 CxF1 3wayFarm
CxF1189308 F 35.0546 CxF1 3wayFarm
```

It has columns individual, sex (which actually has no simulated effect), phenotype, population and farm. Again, it is a single file but it could be split in several files. The number of records per breed is:

Table 2: number of records

line	records
line A	6667
line B	6667
line C	6667
F1	2666
3-way	6667

1.2.3 Markers

We simulated 20 chromosomes of 1M and 2000 markers each. Only 13000 animals are genotyped (2000 for each A, B, C and F1; 5000 for the 3-way cross) - of which 4333 (at random) in data. File `markers` is formatted in the `blupf90` format:

```
  C181153 02202000002020002...
  C181163 12102001001121111...
  C181173 02201100101011112...
AxB181177 12112201211102122...
AxB181181 12112201211102122...
```

of course there are no mistakes in genotyping as this is simulated data. However, probably many markers are monomorphic.

1.3 Models and theory

Now we analyse this with `blupf90test` program (**not** with `blupf90`). The model will include the effects of sex (that it should have no effect, but we pretend that we don't know it yet), animal and farm. We have several options:

Considering a single population. This model assumes that the effect of a gene is the same across different populations ("common genetic" approach (Christensen et al. 2015) or "uniquely defined" effects (Stuber and Cockerham 1966)):

1. BLUP with different means per population, fit as an effect with 5 levels (BLUP).
2. BLUP with unknown parent groups (UPG) for lines A, B and C (BLUP_UPG). This model assumes that the effect of AxB is strictly 1/2 the effect of A and 1/2 the effect of B, and the effect of CxF1 is half the effect of C and 1/4 A and 1/4 B.

These two models are very classic and do not need extra description. The following two including metafounders are more challenging and will be described in the next section:

3. BLUP with metafounders (MF) for lines A, B and C (BLUP_MF). (This model will not be run.)
4. SSGBLUP (including genotypes) with metafounders (MF) for lines A, B and C (SSGBLUP_MF) (Christensen et al. 2015)

We could have considered BLUP (or SSGBLUP) with one random effect per population of origin, i.e. genes have different effects depending whether they come from lines A, B or C. Christensen et al. (Christensen et al. 2015) call this "the partial genetic" approach whereas other people call this “BOA” for “Breed if Origin of Alleles” (Sevillano et al. 2017) whereas Stuber and Cockerham (Stuber and Cockerham 1966) call this “Effects of genes defined according to origin.”[2] Using pedigree (or SSGBLUP), this model is easy to use for F1 crosses (Lo, Fernando, and Grossman 1997 ; Lutaaya et al. 2002). There is no usable correct model with separate pedigrees per breed and crosses beyond F1. Although the theory exists (Lo, Fernando, and Grossman 1993 ; Garcia-Cortes and Toro 2006), this has never been programmed into software. Also the “BOA” model is not more accurate (Xiang, Christensen, and Legarra 2017) than the “common genetic” approach whereas computationally the “common genetic” approach is much simpler.

1.3.1 SSGBLUP with metafounders

The model with MF assumes a relationship within and across founders of the lines. This relationship is matrix $\mathbf{\Gamma}$. This matrix should describe (in our example) that lines A, B and C are related on a distant time and that founders of C in pedigree where actually more related than founders of A. From this matrix, the matrix of pedigree relationships is $\mathbf{A}_{(\Gamma)}$ (Legarra et al. 2015) and its inverse $\mathbf{A}_{(\Gamma)}^{-1}$ is easily obtained. Then, for SSGBLUP we use

$$\mathbf{H}_{(\Gamma)}^{-1} = \mathbf{A}_{(\Gamma)}^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{G}_{05}^{-1} - \mathbf{A}_{22(\mathbf{\Gamma})}^{-1} \end{pmatrix}$$

where a special VanRaden’s $\mathbf{G}_{05}^{-1}$ is constructed using $\mathbf{Z}_{101}$ (genotypes coded as $\{-1, 0, 1\}$) and dividing by $m/2$ with m the number of markers (Garcia-Baccino et al. 2017):

$$\mathbf{G}_{05} = \frac{\mathbf{Z}_{101}\mathbf{Z}_{101}'}{m/2} = 2\frac{\mathbf{Z}_{101}\mathbf{Z}_{101}'}{m}$$

to make invertible $\mathbf{G}$, the software automatically uses bended (or blended) $\mathbf{G}$:

$$\mathbf{G}_{05} = 0.95\frac{\mathbf{Z}_{101}\mathbf{Z}_{101}'}{m/2} + 0.05\mathbf{A}_{22(\mathbf{\Gamma})}$$

but note that *there is no tuning* (usually adjusting means of $\mathbf{G}$ to $\mathbf{A}_{22}$) of $\mathbf{G}_{05}$ - such a compatibility is automatic using metafounders.

1.4 BLUP

1.4.1 Reminder about the two parameter files for `renumf90` and `blupf90`

In the following it is very important to *distinguish the parameter files* for `renumf90` from the parameter file for the other `blupf90` programs (i.e. `blupf90test`, `airemlf90`…). **These are two different files**. Often I use `renum.par` (or similar names e.g. `renum.blup_upg.par`) for the first and `renf90.par` (or similar names e.g. `ren.blupf90.par`) for the second.[3]

1.4.2 Recoding for BLUP using `renumf90`

`renumf90` is a complex but very powerful software that is becoming unavoidable when using the `blupf90` suite. A parameter file indicates where to find the information and what effects to renumber, together with indications on animal effect, pedigree, markers and a few more things. A *very* comprehensive guide of `renumf90` is in Yutaka Masuda’s tutorial. Parameter file `renum.blup.par` contains the instructions for model `BLUP` (I put guesses of variance components):

[2] We geneticists should read more and stop re-inventing the wheel

[3] Perhaps the user could call the `renumf90` parameter files `something.ren` and the `blupf90` parameter files `something.par`.

```
DATAFILE
data
# id sex y breed farm
TRAITS
3
WEIGHT(S)

RESIDUAL_VARIANCE
1
# sex
EFFECT
2 cross alpha
# breed
EFFECT
4 cross alpha
# farm
EFFECT
5 cross alpha
# animal
EFFECT
1 cross alpha
RANDOM
animal
FILE
ped
FILE_POS
1 2 3
PED_DEPTH
3
INBREEDING
pedigree
(CO)VARIANCES
1
```

This is rather self-explicative. The model assumes a single genetic architecture through all animals and is:

$$y = sex + breed + farm + animal + e$$

The only interesting points are:

```
PED_DEPTH
3
```

means that we trace back 3 generations from data and markers. Putting number 100 goes back as far as possible but still prunes animals not ancestors of data or markers - this is useful, for instance, if you have the complete Holstein pedigree but records on 1,000 cows. In our case we go from 89260 animals in the original file `ped` to 35823 (29333 animals with records and their ancestors) in `renadd04.ped`. Putting 0 reads the whole pedigree file.

```
INBREEDING
pedigree
```

computes inbreeding from pedigree and prepares the parameter file `renf90.par` to use inbreeding to create the variance of Mendelian sampling needed to create $\mathbf{A}^{-1}$ through the effect type `add_an_upginb`. We have discovered in the last years that this is important: ignoring inbreeding in the setup of $\mathbf{A}^{-1}$ had usually negligible effects in BLUP, but the consequences in SSGBLUP can be rather dramatic (without inbreeding for $\mathbf{A}^{-1}$, the compatibility among $\mathbf{A}^{-1}$, $\mathbf{G}^{-1}$, and

$\mathbf{A}_{22}^{-1}$ is compromised). Remember `renumf90` does **not** sort pedigrees in birth order. This is how the file `renadd05.ped` looks like:

```
34 30618 6770 2000 0 2 1 0 0 A58455
35 34689 11750 2032 0 2 1 0 0 CxF1192695
30654 30348 30412 2000 0 2 0 0 3 A54485
36 27188 33825 2073 0 2 1 0 0 C179742
```

The file has a series of columns and the 4th column (for instance 2000, 2032… in this example) has the so-called "inb/upg code" that are functions of the variance of mendelian sampling D_{ii}: the inb/upg code has a value of $1000/D_{ii}$ where $D_{ii} = 0.5 - 0.25(F_s + F_d)$ with $F = -1$ for unknown sire or dam. Note that the the inb/upg code does not reflect the inbreeding of an individual but of their parents: for instance CxF1192695 is *not* inbred but one or both parents are (slightly) inbred.

On output, `renf90.par` contains a usable parameter file for the `blupf90` series. This file can be copied with another name (highly recommended), say, `blup.par` as shown in the Annex and then modified by the user.

1.4.3 REML analyses

We can do REML analysis using `blupf90test` adding the following `OPTIONS`:

```
OPTION method VCE
OPTION EM-REML 10
OPTION se_covar_function h2 G_4_4_1_1/(G_4_4_1_1+R_1_1)
```

and in this manner, we can use it in `blupf90test` (that integrates BLUP and REML), to obtain REML estimates of variance components:

```
Final Estimates
 Genetic variance(s) for effect  4
  0.46861
 Residual variance(s)
  0.88372
...
Sampling variances of covariances function of random effects (n=10000)

h2  - Function: g_4_4_1_1/(g_4_4_1_1+r_1_1)
  Mean:   0.34652
  Sample Mean:   0.34632
  Sample SD:   0.13871E-01
```

The latter uses a MonteCarlo method (Houle and Meyer 2015) and for technical reasons may not give a "Sample Mean" equal to the "Mean". Anyway, the "Mean" is the value to be trusted.

The estimate of h^2 is not 0.5 but 0.35 as the variances are heterogeneous across breeds. Later we will show a 5-trait model where this is considered.

1.4.4 Solutions for breed and farm effects.

With these estimated values we can run BLUP analyses. The file `solutions` contains estimates of effects, but with recoded numbers - one needs to go back to original codes in `renf90.tables`. After doing that, we obtain:

```
F        0.97861500E-02
M         0.0000000
A         27.611539
AxB         30.195616
B        13.494751
```

```
C        28.596234
CxF1        45.471410
3wayFarm        0.0000000
F1Farm        0.0000000
PurebredFarm        0.0000000
```

Farm and breed effects are confounded and the software assigns 0 value to breeds. If you go to Table 1 you will find that "breed" results agree well with the averages for y, but not with the average genetic values - it does not separate breed genetic effects from environmental effects.

1.4.5 Other solutions for breed and farm effects

This is another solution obtained with an iterative method instead of direct inversion:

```
1 1 F        9.6418269
1 2 M        9.6320436
2 1 A        11.128977
2 2 AxB        10.277400
2 3 B       -2.8984491
2 4 C        12.153085
2 5 CxF1        17.909376
3 1 3wayFarm        17.909376
3 2 F1Farm        10.277400
3 3 PurebredFarm        6.7945374
```

The solution seems radically different from the previous one, but it is not. Both are valid solutions. Because the model is not full rank, only estimable functions are estimable. Because farms and breeds are confounded, all that we can say is that, averaging both sexes $((9.64 + 9.63)/2)$, the expected phenotype in the *3wayfarm* (estimated effect 17.90) with *CxF1* (estimated effect *also* 17.90) animals is $(9.64 + 9.63)/2 + 17.90 + 17.90 = 45.43$, identical to the previous result (minus some rounding error). Thus, the fact that we get numbers for Farms and Breeds is misleading - we cannot estimate them jointly with this model.

1.5 BLUP_UPG

1.5.1 Parameter file for BLUP_UPG

In this case we will *not* fit an explicit "breed" effect - we fit Genetic or Unknown Parent Groups (Quaas 1988).

1.5.1.1 Pruning pedigrees and UPG renumf90 has a few options to create Genetic Groups, also known as Unknown Parent Groups (UPG). One of them allows simply to assign each animal to some "population", and if parents are unknown **or** parents are beyond the number of generation traced back, it assigns UPGs based on this. The code for this in the parameter file is similar as before:

```
FILE_POS
1 2 3 0 0 4
PED_DEPTH
3
UPG_TYPE
group_unisex
INBREEDING
pedigree
```

The UPG_TYPE option group_unisex takes a column in pedigree file to assign UPG - the same for the sire and for the dam.[4] This column is indicated with the *6th number* in the line 1 2 3 0 0 4 located under FILE_POS - in this case, the 4th column. Remember that our pedigree file looked like:

```
A1766 A6 A51 A
A1767 A6 A51 A
```

so, it is going to pick “A” (4th column) as identifier of UPG. Using this we get in the output:

```
Unknown parent group allocation
       OriginalGroup   Group       UPGID      #Animals
                 C       1       35824           840
                 A       2       35825           840
                 B       3       35826           840
Wrote UPG file "renf90.upg" with original id
Max group = 3; Max UPG ID = 35826
```

Why don’t we have AxB and CxF1 as UPGs? because pedigrees trace back eventually to A,B,C.

```
Unknown parent group allocation
        Parent2         C        A        B
Parent1     34563         0        0        0
C               0       420        0        0
A               0         0      420        0
B               0         0        0      420
```

means that 34563 individuals have known father and sire, but 420 of each pure breed do not have neither. Now the pedigree file renadd03.ped has no zeros on it - but UPGs do not have lines on their own.

The REML analyses gives exactly the same results as above.

1.5.2 Solutions for breed effects and UPG effects.

After REML, solutions for farm effects are:

```
3wayFarm         34.391471
F1Farm        23.137222
PurebredFarm         13.494751
```

and for UPGs:

```
C         15.101483
A         14.116788
B         0.0000000
```

Solutions for sex are 0. These results agree quite well with Table 1 (beyond a constant) and the UPG model separates correctly genetic and phenotypic values of breeds and farms. The reason is, in a way, that breeds A and B “connect” all three different farms. This of course assumes no strong GxE interactions.

1.6 SSGBLUP_MF

However, breed composition is *not* the same for 3-way crosses. The mother of a CxF1 animal has one gamete (chromosome) from A and another from B, and it may transmit one, the other, or a recombined chromosome to the CxF1 offspring. Thus, on the maternal side, some CxF1 animals

[4] Another strategy (group) more used in cattle and sheep assigns different groups for sire and dams, but this is delicate as it is easy to have confounded effects. For purebreds and crossbreds I recommend to assign UPGs based on breed or line.

will be more “A” and some will be more “B”. This is the kind of information that can be “seen” using molecular markers. We will not enter into the theory, but this is the kind of idea that is well included into the theory of metafounders.

1.6.1 Parameter file for SSGBLUP_MF

We need to prepare the files partly automatically and partly by hand. We use parameter file `renum.ssgblup.par` that is almost identical to `renum.blup_upg.par` except for these two lines (usually placed after `FILE_POS`):

```
SNP_FILE
markers
```

which tells `renumf90` to pick up the genotype file and animals in there. This generates a parameter file `renf90.par` to run SSGBLUP with UPGs. **This is not a good model** because genomic information is not accounted for in the UPG effects (Misztal et al. 2013 ; Matilainen et al. 2018). Two alternatives are to use the QP transformation into the “genomic” part of the model, so-called “exact UPGs” (we will not use this option as it is more complicated), or using metafounders. To use metafounders we need:

1. to modify the parameter file
2. to provide the relationship matrix across metafounders, $\mathbf{\Gamma}$.

1.6.2 Preparing the `blupf90` parameter file to consider metafounders

After running `renumf90`, in the parameter file for `blupf90 family` (`renf90.par` or another file name if you copied it) we need to change the type of animal effect (`RANDOM_TYPE`) from `add_an_upginb` to `add_an_meta`. This tells `blupf90test` that the random effect follows the structure of $\mathbf{A}^{\mathbf{\Gamma}}$.

We also need to provide the software with an estimate of $\mathbf{\Gamma}$, and we do this using the line `OPTION gammafile 3 name_of_the_file` where 3 is the random effect associated with model and `name_of_the_file` is the file with the values for $\mathbf{\Gamma}$, which looks like:

```
0.799   0.456   0.456
0.456   0.661   0.454
0.456   0.454   0.670
```

ordered as in the output file `renf90.upg` of renumf90 (in our case, the order is C, A and B). We will talk about getting these values later. So far, this tells the software to do (regular) BLUP with $\mathbf{A}^{\mathbf{\Gamma}}$. When we use this relationship matrix, the software automatically scales the variance component appropriately (Legarra et al. 2015) such that

$$\sigma_u^2 \leftarrow \frac{\sigma_u^2}{1+\overline{diag(\mathbf{\Gamma})}/2-\overline{\mathbf{\Gamma}}}$$

This scaling factor results in an *increase* of the nominal genetic variance. The automatic scaling can be deactivated with `OPTION lscale F`.[5]

If we, in addition, use the option `OPTION SNP_file name_of_the_file` (this is by default already present in the parameter file as constructed by `renumf90`), the program will run SSGBLUP with

$$\mathbf{H}_{(\Gamma)}^{-1} = \mathbf{A}_{(\Gamma)}^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{G}_{05}^{-1} - \mathbf{A}_{22(\mathbf{\Gamma})}^{-1} \end{pmatrix}$$

This is how the parameter file for `blupf90test` for SSGBLUP using metafounders looks like:

[5] The scaling assumes, roughly, that all metafounders contribute equally to the overall population - it may not be correct if they are very different and one contributes very little.

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

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         2 cross
 3         3 cross
 4     43750 cross
RANDOM_RESIDUAL VALUES
   1.0000
 RANDOM_GROUP
     3
 RANDOM_TYPE
 add_an_meta
 FILE
renadd03.ped
(CO)VARIANCES
   1.0000
OPTION SNP_file markers
OPTION gammafile 3 gamma.txt
```

Adding the following lines results in REML estimation, in addition using YAMS (faster) and with s.e. for heritability:

```
OPTION use_yams
OPTION method VCE
OPTION EM-REML 10
OPTION se_covar_function h2 G_3_3_1_1/(G_3_3_1_1+R_1_1)
```

Running the REML analysis shows at the beginning lots of information, including the scaling factor $k = 1 + \overline{diag(\mathbf{\Gamma})}/2 - \overline{\mathbf{\Gamma}}$ which in this case is 0.8148:

```
* Scale_k =  0.8148
* Matrix G (genetic covariances) is scaled with k=  0.8148
 Original G
  1.000
 Scaled G
  1.227
```

The estimate of variance components at the end is

```
Genetic variance(s) for effect  3
  1.0270
Residual variance(s)
 0.71508
```

However the estimate of genetic variance is 1.0270 in the *metafounders* scale, to put it back in the *normal* scale we multiply by $k = 1 + \overline{diag(\mathbf{\Gamma})}/2 - \overline{\mathbf{\Gamma}} = 0.8148$ in this case which results in $\widehat{\sigma_u^2} = 1.0270 \times 0.8148 = 0.84$. These numbers are anyway not completely meaningful as there are scaling effects across all 5 breeds. All these things of the variances are confusing, but in practice the automatic scaling works quite well in our experience.

1.6.3 Solutions for breed effects and UPG effects.

After REML, solutions for farm effects are:

```
 3wayFarm        20.827173
 F1Farm        9.6071542
 PurebredFarm        0.0000000
```

which agree very well with simulated data, and for metafounders:

```
C         6.2802710
A         -8.2012730
B         5.2292554
```

which also agree very well.

1.7 Inferring Gamma

1.7.1 Using home-made breed allele frequencies

Each element γ_{ij} of matrix $\mathbf{\Gamma}$ contains $8Cov(p_i, p_j)$ where i and j are the allele frequencies at the top of the pedigree of each breed (i.e. we have 40,000 allele frequencies for each A, B and C). It is important that the reference allele is randomized so on average, $\bar{p} = 0.5$ across loci.

To compute allele frequency at the base population we can pick up the first 1000 animals of each breed in `markers` and get allele frequencies using some program. I use my own programs and I get 3 files, `freqA`, `freqB`, `freqC`, each of them looking like:

```
0.43
0.936
...
0.092
```

from here, we use R. The order given by `renumf90` was C, A, B so I use the same order to generate my estimate of $\mathbf{\Gamma}$:

```
freqA=scan("freqA")
freqB=scan("freqB")
freqC=scan("freqC")
8*cov(cbind(freqC,freqA,freqB))
```

which gives

```
          freqC     freqA     freqB
freqC 0.8108728 0.4614458 0.4602379
freqA 0.4614458 0.6823929 0.4585665
freqB 0.4602379 0.4585665 0.6723112
```

which is very similar to the values that we described before. Paste this (without row and column names) in a text file and include the name of this file in `OPTION gammafile`.

1.7.2 using gammaf90

`Gammaf90` is a software still unreleased that uses the GLS method of (Garcia-Baccino et al. 2017), and is most useful when there is no genotypes of "purebred" animals, only of their crosses - for instance if we only had genotypes of F1 and 3way crosses. `Gammaf90` has *no* parameter file, but command switches. In our case is as simple as:

```
gammaf90 --snpfile markers --pedfile renadd03.ped
  --allele-freqs-base --bounded-allele-freqs
```

The switch `--allele-freqs-base` creates a file with estimated base allele frequencies for each metafounder, and approximate s.e. As estimates of allele frequencies may go out of $[0, 1]$ bounds, the switch `--bounded-allele-freqs` prevents them by fixing them brutally to the $[0, 1]$ bound - not very elegant but useful.

On output we have `gamma.txt`. In this simulation the estimate is very easy - but in other cases such as ruminants with unknown pedigrees it is much more challenging.

1.8 Five trait model

Because of the existence of GxG and GxE interaction, treating each breed composition as a different "trait" is a typical option for analysis. In this case, we need to prepare the data file in five columns.

2 Annex

Contents of the folder:

- `markers.gz` (compressed file)
- `data`
- `ped`

In folder `renfiles` we have example files for `renumf90`:

- `renum.blup.par`
- `renum.blup_upgs.par`
- `renum.ssgblup.par`

In folder `parfiles` we have example files for `blupf90test`:

- `blup.par`
- `blup_upgs.par`
- `ssgblup.par`

References

Christensen, Ole F., Andres Legarra, Mogens S. Lund, and Guosheng Su. 2015. "Genetic Evaluation for Three-Way Crossbreeding." *Genetics Selection Evolution* 47 (1): 98. http://link.springer.com/article/10.1186/s12711-015-0177-6.

Garcia-Baccino, Carolina A., Andres Legarra, Ole F. Christensen, Ignacy Misztal, Ivan Pocrnic, Zulma G. Vitezica, and Rodolfo J. C. Cantet. 2017. "Metafounders Are Related to Fst Fixation Indices and Reduce Bias in Single-Step Genomic Evaluations." *Genetics Selection Evolution* 49: 34. https://doi.org/10.1186/s12711-017-0309-2.

Garcia-Cortes, L. A., and MA Toro. 2006. "Multibreed Analysis by Splitting the Breeding Values." *Genetics Selection Evolution* 38 (6): 601–15.

Houle, D., and K. Meyer. 2015. "Estimating Sampling Error of Evolutionary Statistics Based on Genetic Covariance Matrices Using Maximum Likelihood." *Journal of Evolutionary Biology* 28 (8): 1542–9. https://doi.org/10.1111/jeb.12674.

Legarra, Andres, Ole F. Christensen, Zulma G. Vitezica, Ignacio Aguilar, and Ignacy Misztal. 2015. "Ancestral Relationships Using Metafounders: Finite Ancestral Populations and Across Population Relationships." *Genetics* 200 (2): 455–68. https://doi.org/10.1534/genetics.115.177014.

Lo, LL, RL Fernando, and M. Grossman. 1997. "Genetic Evaluation by BLUP in Two-Breed Terminal Crossbreeding Systems Under Dominance." *Journal of Animal Science* 75 (11): 2877–84.

Lo, L. L., R. L. Fernando, and M. Grossman. 1993. "Covariance Between Relatives in Multibreed Populations - Additive-Model." *Theoretical and Applied Genetics* 87 (4): 423–30.

Lutaaya, E., I. Misztal, JW Mabry, T. Short, HH Timm, and R. Holzbauer. 2002. "Joint Evaluation of Purebreds and Crossbreds in Swine." *Journal of Animal Science* 80 (9): 2263–6.

Matilainen, Kaarina, Ismo Strandén, Gert Pedersen Aamand, and Esa A. Mäntysaari. 2018. "Single Step Genomic Evaluation for Female Fertility in Nordic Red Dairy Cattle." *Journal of Animal Breeding and Genetics* 135 (5): 337–48. https://doi.org/10.1111/jbg.12353.

Misztal, I., Z. G. Vitezica, A. Legarra, I. Aguilar, and A. A. Swan. 2013. "Unknown-Parent Groups in Single-Step Genomic Evaluation." *J Anim Breed Genet* 130. https://doi.org/10.1111/jbg.12025.

Quaas, R. L. 1988. "Additive Genetic Model with Groups and Relationships." *Journal of Dairy Science* 71: 1338–45.

Sargolzaei, Mehdi, and Flavio S Schenkel. 2009. "QMSim: A Large-Scale Genome Simulator for Livestock." *Bioinformatics* 25 (5): 680–81. http://dx.doi.org/10.1093/bioinformatics/btp045.

Sevillano, Claudia A., Jeremie Vandenplas, John W. M. Bastiaansen, Rob Bergsma, and Mario P. L. Calus. 2017. "Genomic Evaluation for a Three-Way Crossbreeding System Considering Breed-of-Origin of Alleles." *Genetics Selection Evolution* 49 (1): 75. https://doi.org/10.1186/s12711-017-0350-1.

Stuber, C. W., and C. Clark Cockerham. 1966. "Gene Effects and Variances in Hybrid Populations." *Genetics* 54 (6): 1279–86. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1211293/.

Xiang, Tao, Ole Fredslund Christensen, and Andres Legarra. 2017. "Technical Note: Genomic Evaluation for Crossbred Performance in a Single-Step Approach with Metafounders." *Journal of Animal Science* 95 (4): 1472–80.