Česká plemenářská inspekce

Štěpánská 626/63, Praha 1, 11000

v y d á v á

O S V Ě D Č E N Í

(8498/2014 - ČPI)

o uznání uplatněné certifikované metodiky
v souladu s podmínkami „Metodiky hodnocení výsledků výzkumu a vývoje“

Genetic evaluation by linear models using own algorithms and standard software

Autor a předkládající organizace: J. Přibyl, J. Bauer, F. Krupa, Z. Krupová, M. Milerski, A. Novotná, P. Pešek, J. Přibylová, J. Schmidová, A. Svitáková, Z. Veselá, H. Vostrá Vydrová, L. Vostrý, L. Zavadilová, E. Žáková, Výzkumný ústav živočišné výroby, v. v. i. Přátelství 815, 104 00 Praha Uhříněves.

Vypracované v rámci výzkumného projektu QI111A167.

V Praze dne 14.10.2014

Česká plemenářská inspekce
Štěpánská 626/63
110 00 Praha 1
1

(Razítko odborného orgánu státní správy):

(Jméno a funkce zástupce odborného útvaru státní správy): ing. Zdenka Majzlíková
ředitelka

(Podpis zástupce odborného útvaru státní správy): .....

INSTITUTE OF ANIMAL SCIENCE
PRAHA UHŘÍNĚVES, CZECH REPUBLIC

CERTIFIED METHODOLOGY

GENETIC EVALUATION BY LINEAR MODELS USING OWN ALGORITHMS AND STANDARD SOFTWARE

Authors

J. Přibyl, J. Bauer, E. Krupa, Z. Krupová, M. Milerski, A. Novotná, P. Pešek, J. Přibylová, J. Schmidová, A. Svitáková, Z. Veselá, H. Vostrá Vydrová, L. Vostrý, L. Zavadilová, E. Žáková

Opponents

Ing. Zdenka Majzlíková
Česká plemenářská inspekce, Praha

Ing. Jiří Šplíchal
Českomoravská společnost chovatelů, Hradištko

Elaborated with support by the Czech Ministry of Agriculture,
Project QI111A167 (Genomic selection of dairy cattle).

2014

ISBN: **978-80-7403-128-1**

CONTENTS

I. Objective of methodology

Objective of the methodology is a short survey of methodology of linear models used for genetic evaluation of large data sets of animals (until millions and more of equations) and application of this methodology for different kinds of data according the nature of a trait. Focus is in practical application using own programming and using software accessible on the internet. Users of methodology are persons working in nation-wide evaluation of animals and scientists. Could be used also for teaching of students at universities.

II. Description of methodology

1. Introduction

Presented text covers introduction into theory of linear models and basic methodology used in genetic evaluation. Traditional pedigree based approaches and approaches exploiting huge number of markers from genetic chips are demonstrated. Manual leads the reader from simple examples to complex procedures by his own active work with computer and by studying algorithms for different calculations. The working tool is programming in SAS, mainly in matrix algebra IML. Similar practise could be used with other programming environment. The text has two principal parts:

- Constructing and solving systems of equations for genetic predictions in matrix algebra.
- Introduction to free available software: BLUPF90-family and DMU.

Covers topics of:

- Matrix algebra in The SAS, solving system of equations, transforming data-files into matrices
- Derivation of linear model
- Construction of matrices design for independent variables effects
- Construction of the system of normal equations
- Least-square method (LSM)
- Construction of numerator relationship matrix **A**
- Direct construction of inverse of **A**
- BLUP - Animal Model
- Construction of matrix of genetic markers
- Prediction of regression coefficients by ridge regression method RRBLUP
- Calculation of direct genetic value DGV
- Construction of genomic relationship matrix **G**
- Calculation of DGV by GBLUP
- Augmenting **A** by **G**
- Prediction of genomic enhanced breeding value (GEBV) by single-step procedure ssGBLUP
- Introduction to use of BLUPF90-family programs
- Introduction to use of DMU programs

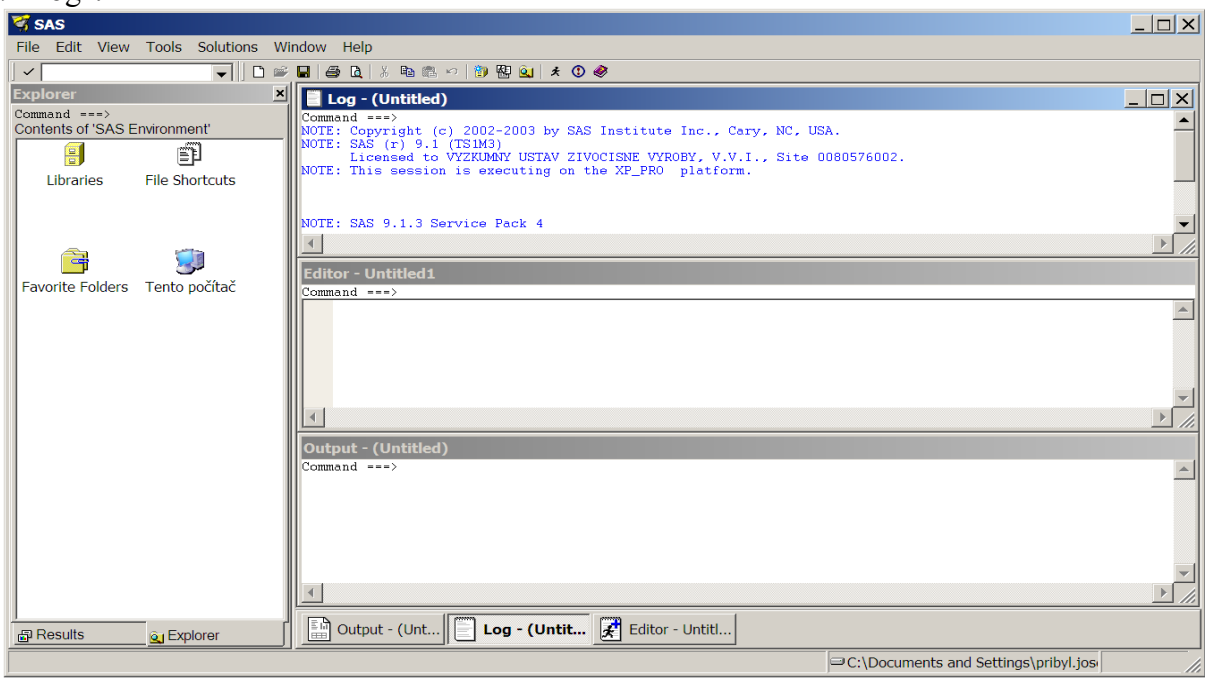
Prerequisite for user is basic knowledge of matrix algebra and basic use of computers. Active programming on computer is necessary, therefore previous knowledge of construction and application of algorithms is useful.

Cited programs and study materials are used wit kind permission of authors Dr. Per Madsen, Prof. Ignacy Misztal, Dr. Larry Schaeffer and SAS representative in the Czech Rep.

2. Writing programs for matrix algebra and files in The SAS

Licence for universities allows to install The SAS software freely on student's computers or laptops. The SAS software is composed by several modules. We are using the "Base" module, which is efficient for handling data files and module "Interactive matrix language (IML)", which allows easy manipulation in matrix algebra.

After starting the The SAS, the leading screen contains three main windows "Editor", "Output" and "Log".



"**Editor**" is used to write the text, create your own programs and edit them.

"**Output**" contains results.

"**Log**" contains messages about processing of your program, including warnings and errors.

Into editor window could be **imported** files from your directory on drive. Content of all three windows could be separately **exported** (saved) to your directory, or printed.

Submit the program: to run the whole code in editor window, just click on "figure" icon on main task bar or type "F8" on keyboard. To run only part of code (f.e. some procedures only) highlight it first by mouse and then click on "figure" icon or type "F8". You have to have active cursor in Editor window.

Clean the content of a window, where you have active cursor, by clicking on blank page bottom in main taskbar, or by simultaneous pressing " Ctrl " + " e ".

Recalling content of window by " F4 " button.

In a main taskbar is button for help.

- You can locate **multi line comments** to any place in your program within marks: /* .. comment...*/, for comment a single line (terminated by semicolon) just write *.
- Each command is terminated by **semicolon " ; ".**
- Items in a command are **separated by space**. Number of spaces is not important.
- In one row could be several commands. One command could be in several lines.
- The code is not case sensitive.

Next examples can be copied directly into Editor window in The SAS and submitted to run. Attached files must be copied into your directory and statements "filename" in examples must be modified according to the path of your directory. Examples are connected with directory c:/LinMod/myprog/.

Matrix operations

```
Proc IML ;              /* calling the procedure IML */
reset print ;           /* instruction to print all */
A =  {  2 1 1 ,        /* creating matrix A */
        0 1 3 } ;
B =   { 1 1 2 ,
        1 1 0 } ;       * creating matrix B ;
C = a + b ;             /* summing A with B */
D = a*b` ;              /* matrix multiplication A with transposition of B */
e  = a#b ;              /* elements multiplication */
f  = inv(d) ;           /* inverse of D */
g = block(d,f) ;        /* combine matrices D and F diagonally */
h = A||B ;              /* horizontal concatenation of matrices A and B*/
q  = a//b ;             /* vertical concatenation of matrices A and B */
r = diag(d) ;           /* create diagonal matrix from D */
k = vecdiag(d) ;        /* move diagonal to vector */
l  = j(3,4,7) ;         /* create matrix with 3 rows and 4 columns of identical values 7 */
m = i(3) ;              /* create identity matrix of size 3 */
n = q[2:3,1:2] ;        /* creation matrix N by cutting out from matrix Q */
n[1,1] = 8 ;            /* rewriting the given element of N by the value 8 */
o  = nrow(a) ;          /* calculation the number of rows in A */
p  = trace(d) ;         /* calculation trace of matrix D */
quit ;                  /* termination with IML */
```

Work with files

Each program must be **terminated** by command **“run ; “** except procedures iml, sql, gplot and gchart which are **terminated** by **“quit;”**.

```
/*  .............              myfiles          ..................*/
/* .........  some basic operation with files  ..............  */
filename   prod     "c:\LinMod\myprog\uzit" ;    /* localization of input file */
filename   prodcor "c:\LinMod\myprog\uzitcor";/* localization of output file */
data a;
     infile prod ;                          /* reading file from drive */
     input milk animal herd age do ;        /* variables in input file separated by space */
title " File a";                            /* printing of title */
proc means ;                                /* descriptive statistics */
proc freq ; tables herd ;                   /* frequency table according to herds */
proc univariate data=a normal plot;         /* print the distribution of variable milk */
     var milk ;
proc print data=a ;                         /* print of data A */
data b ;
     set a ;                                /* insert data A into data B */
     if herd > 1 then delete ;              /* eliminate herds except herd = 1 */
     drop milk herd ;                       /* eliminate variable milk and herd */
     agec = age -27 ;                       /* subtract average of age at calving */
     doc   = do - 90 ;                      /* subtract average of days open */
title " File b";
proc means ;
```

```
data c ;
      set a ;
      if herd  ne 2 then delete ;              /* eliminate herds except herd = 2 */
      keep animal age do agec doc;             /* keep only mentioned variables */
      agec = age -27 ;
      doc  = do - 90 ;
title " File c";
proc means ;
data d ;
      set b  c ;                               /* insert data B and below data C into data D */
title " File d";
proc means ;
proc sort data=a ; by animal ;                 /* sorting according to animals */
proc sort data=d ; by animal ;
data e ;
      merge a d ; by animal ;      /* merging side by side files A and D according to animals */
      age2 = age*age ;             /* creation variable with second power of age */
title " File e";
proc means ;
data f ;
      set e ;
      if agec = . then delete ;    /* when variable agec is missing, then eliminate observation */
      file  prodcor ;              /* writing the file to drive */
      put milk animal herd age agec age2 do doc ;
title " File f";
proc means ;
run ; /*.....finish.....*/
```

Manipulation with files and matrices

```
/*  .....                file-mat          .....*/
/* .....  files into matrices and contrary ..... */
filename   prod  "c:\LinMod\myprog\uzit" ;   /* localization of input file */
filename   vey   "c:\LinMod\myprog\vey" ;    /* localization of output file */
                /*..... file .....*/
data a;
      infile prod ;
      input milk animal herd age sp ;
      keep milk ;
proc means ;
               /*..... matrices .....*/
proc IML ;
      use a ;
      read all into ml ;        /* converting milk from file A into vector ML */
      close a ;
      ss = ml`*ml ;             /* calculation the sum of squares */
      print ss ;                /* print ss */
      create y from ml ;        /* vector ML into file Y */
      append from ml ;
              /*..... file .....*/
data b ;
```

```
    set y ;
    mlk = col1 ;          /* column 1 from matrix into variable mlk */
    file vey ;            /* writing the file to drive */
    put mlk ;
proc means ;
run ; /*.....*/
```

Some basic SAS tutorials:
http://www.yorku.ca/pek/index_files/quickstart/IMLQuickStart.pdf
https://support.sas.com/resources/papers/proceedings13/144-2013.pdf
http://blogs.sas.com/content/iml/files/2011/10/IMLTipSheet.pdf
http://blogs.sas.com/content/iml/2011/10/10/sasiml-tip-sheets/
http://support.sas.com/rnd/app/video/index.html#iml

3. Linear model with fixed effects

Data are frequently evaluated by linear models, which are explained in many handbooks and manuals, for example Přibyl and Přibylová (2002), Mrode (2014) and Schaeffer (2014). Overview of methodology useful for genomic evaluation is for example in Přibyl et al. (2010) and Legarra et al. (2014).
Linear model with fixed effect is possible to described by model equation

$$\mathbf{Y} = \mathbf{Xb} + \mathbf{e} \quad , \qquad (1)$$

Where: **Y** is known vector of observed values, *dependent variable*,
X is known design matrix of plan of experiment connecting observations in **Y** with estimated parameters in **b**,
b is unknown vector of levels of estimated effects, *independent variable*,
e is unknown vector of random errors, with a residual variance σ^2_e.

Matrices could be subdivided into blocks for more effects and simultaneously evaluate more traits (Multi-Trait (MT)). In a case of MT, σ^2_e is substituted by residual covariance matrix between traits **R0**.

Finesse of breeders and researchers is to propose the model which has the smallest random error and most reliable estimation of **b**. Estimation of **b** is therefore not arbitrary, and must be optimised, frequently by finding the minimum (extreme) of the loss function:

$$lf = \hat{e}'\hat{e} = (Y - X\hat{b})'(Y - X\hat{b}) \qquad (2)$$

Finding the extreme of function is done by partial differentiations of lf according to elements (i) of **b** and putting them equal to zero

$$\frac{\partial lf}{\partial b_i} = 0_i \quad , \qquad (3)$$

By the algebraic rearrangement of the system of all these equations according (i) we will receive the system of normal equations

$$\left[X'R^{-1}X\right]\left[b\right] = \left[X'R^{-1}Y\right] \quad , \qquad (4)$$

where **R** is a residual covariance matrix of random errors. Random errors are frequently independent and then **R** becomes diagonal, only with variances of elements of **e**, or diagonal blocs **R0**. Values on diagonal of $\mathbf{R}^{-1}$ can be considered as weights of observations.

Inversion of left hand size (LHS) is

$$[C] = [X' R^{-1} X]^{-1} \quad , \qquad (5)$$

and solution for **b** yields from matrix multiplication of the inverse of (LHS) with right hand side (RHS)

$$[\hat{b}] = [C].[X' R^{-1} Y] \quad , \qquad (6)$$

When in a Single-Trait (ST) all observations in **Y** have the same error, it also means the same weights, $\mathbf{R}^{-1}$ is possible from (4) to cancel out, and the system of equations becomes

$$[X'X].[b] = [X'Y] \quad , \qquad (7)$$

Technique based on (7) is Least Square Method (LSM) and based on (4) Generalised Least Squared Method (GLSM). GLMS is suitable for weighted analysis, when different observations have different weight.

In following examples we are using for simplicity inversion of LHS matrix for solution. In real life systems of equations are huge and iterative procedures are applied.

Example 1. Simple average

We have 5 observations of milk, suppose that all of them are recorded with the same error. We do not have more information. Only what we can do, is to estimate one parameter in **b**, which is in this case the mean. Matrix **X** has one column. Solution is according to (7).

```
proc IML ; reset print ;
    y = { 7000 , 8000 , 6000 , 9000 , 8000 } ;
    x = { 1 , 1 , 1 , 1 , 1 } ;     /* experiment design for one group */
    xx = x`*x ;                     /* left hand side  LHS */
    xy = x`*y ;                     /* right hand side RHS */
    c = inv(xx) ;                   /* inversion of LHS */
    b = c*xy ;                      /* solution */
quit;
```

Example 2. Two herds

Like (Ex.1), but now we know that first three observations are from herd 1 and last two from herd 2. We can compare averages of herds. We are working with an effect **b**, which has 2 classes, therefore design matrix **X** has two columns connecting observations **Y** with herd 1 and herd 2.

```
proc IML ; reset print ;
    y = { 7000 , 8000 , 6000 , 9000 , 8000 } ;
    x = { 1  0 ,                /* design for herds */
          1  0 ,
          1  0 ,
          0  1 ,
          0  1 } ;
    xx = x`*x ;
    xy = x`*y ;
    c = inv(xx) ;
    b = c*xy ;   differen = b[1] - b[2] ;
quit;
```

Example 3. Herds and regression

Like (Ex.2), but we received additional information about the age at first calving. Age of calving is the continuous variable (not in classes) and we will estimate the regression coefficient for this covariable. Matrix **X** and vector **b** have now two parts, $\mathbf{X_1}$ and $\mathbf{X_2}$ and $\mathbf{b_1}$ and $\mathbf{b_2}$ for herds and age. Calculation can be done with "entire" matrix **X** containing both $\mathbf{X_1}$ and $\mathbf{X_2}$ ($\mathbf{X} = \mathbf{X_1} \parallel \mathbf{X_2}$), or the system of normal equations (7) can be modified into:

$$\begin{bmatrix} X_1'X_1 & X_1'X_2 \\ X_2'X_1 & X_2'X_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} X_1'Y \\ X_2'Y \end{bmatrix}, \qquad (8)$$

Compare results of example 2 and example 3.

```
proc IML ; reset print ;
     y = { 7000 , 8000 , 6000 , 9000 , 8000 } ;
     x1 = {  1  0 ,
             1  0 ,
             1  0 ,
             0  1 ,
             0  1 } ;
     x2 = { 27 , 28 , 27 , 28 , 28 } ;
     x2 = x2 - 27 ;                  /* standardization of age to 27 months */
     x1x1 = x1`*x1 ;  x1x2 = x1`*x2 ;
     x2x1 = x1x2` ;   x2x2 = x2`*x2 ;
     r1 = x1x1||x1x2 ;               /* creation of LHS */
     r2 = x2x1||x2x2 ;
     lhs = r1//r2 ;
     x1y = x1`*y ;                   /* creation of RHS */
     x2y = x2`*y ;
     rhs = x1y//x2y ;
     c = inv(lhs) ;
     b = c*rhs ;                     /* solution */
    differen = b[1] - b[2] ;         /* difference between herds */
quit;
```

Example 4. More cross-classified effects

Like (Ex.3), but observations 2 and 4 are breed 1, others are breed 2. The system of equations is extending to 3 effects. Two cross-classified effects in classes (1 and 3) produce the dependency of equations (sum of equations for herds and sum of equations for breeds are the same). Therefore system of equations has not solution. Condition of solvability has to be added to the system of equations for each addition fixed effect in classes. We use the condition that the breed 1 is a base (breed1 = 0) and breed 2 will be expressed as deviation from this base.

```
proc IML ; reset print ;
     y = { 7000 , 8000 , 6000 , 9000 , 8000 } ;
     x1 = { 1  0 ,
            1  0 ,
            1  0 ,
            0  1 ,
            0  1 } ;
     x2 = { 27 , 28 , 27 , 28 , 28 } ;
     x2 = x2 - 27 ;
     x3 = {0  1 ,                    /* design for breeds */
```

```
        1  0 ,
        0  1 ,
        1  0 ,
        0  1 } ;
    x1x1 = x1`*x1 ;  x1x2 = x1`*x2 ;  x1x3 = x1`*x3 ;
    x2x1 = x1x2` ;   x2x2 = x2`*x2 ;  x2x3 = x2`*x3 ;
    x3x1 = x1x3` ;   x3x2 = x2x3` ;   x3x3 = x3`*x3 ;
    r1 = x1x1||x1x2||x1x3 ;          /* creation of LHS */
    r2 = x2x1||x2x2||x2x3 ;
    r3 = x3x1||x3x2||x3x3 ;
    lhs = r1//r2//r3 ;
    x1y = x1`*y ;                    /* creation of RHS */
    x2y = x2`*y ;
    x3y = x3`*y ;
    rhs = x1y//x2y//x3y ;
    condc = {0 , 0 , 0 , 1 , 0 } ;   /*column conditions of solvability position of  "breed1"*/
    lhs = lhs||condc ;
    condr = {0  0  0  1  0  0 } ;    /* row conditions of solvability */
    lhs = lhs//condr ;
    rhs = rhs//0 ;                   /* breed1 = 0 */
    c = inv(lhs) ;
    b = c*rhs ;
    differen = b[1] - b[2] ;
quit;
```

4. Linears models with random (animal) effect

BLUP – Animal Model

When in the model are random effects,

$$\boldsymbol{Y} = \boldsymbol{Xb} + \boldsymbol{Zu} + \boldsymbol{e} \quad , \qquad (9)$$

where **Z** is known design matrix of plan of experiment of random effect connecting observations in **Y** with predicted parameters in **u**,

u is unknown vector of levels of predicted random effects (breeding values), independent variable with a variance σ^2_u.

Prior variance components are included and the system of equations with several effects (8) will change to

$$\begin{bmatrix} X'R^{-1}X & X'R^{-1}Z \\ Z'R^{-1}X & Z'R^{-1}Z + M^{-1} \end{bmatrix} \cdot \begin{bmatrix} b \\ u \end{bmatrix} = \begin{bmatrix} X'R^{-1}Y \\ Z'R^{-1}Y \end{bmatrix} \qquad (10)$$

where $\mathbf{M} = \mathbf{A} \otimes \sigma^2_u$, (11)

$\otimes$ is direct (Kronecker) multiplication of matrices,

A is matrix, which express the dependency between levels of random effect (numerator relationship matrix).

When the random effect is only one and genetic, the sum of $\sigma^2_u + \sigma^2_e$ is the phenotype variance:

$\sigma^2_P = \sigma^2_u + \sigma^2_e$

$h^2 = \sigma^2_u / \sigma^2_P$

In the MT analysis, σ^2_u and σ^2_e are substituted by covariance matrices **G0** and **R0**.

Let the simple example with only one random animal effect and one trait, constant and independent residuals, then the system (10) could by analogically to system (7) simplified into:

$$\begin{bmatrix} X'X & X'Z \\ Z'X & Z'Z + \lambda A^{-1} \end{bmatrix} \cdot \begin{bmatrix} b \\ u \end{bmatrix} = \begin{bmatrix} X'Y \\ Z'Y \end{bmatrix} \qquad (12)$$

where $\lambda = \sigma^2_e / \sigma^2_u = (1 - h^2) / h^2$, (13)

Example 5. ST animal model

Like (Ex.3), but observations are cows. System is extended to 3 effects, two fixed + one random animal. Heritability is 0.30. We have not information about relationship, therefore relationship matrix **A** is the identity matrix. Solution is done according to (12). Which cow is the best?

```
proc IML ; reset print ;
    h2 = 0.30 ;
    lamb = (1-h2)/h2 ;
    y = { 7000 , 8000 , 6000 , 9000 , 8000 } ;
    x1 = { 1  0 ,
           1  0 ,
           1  0 ,
           0  1 ,
           0  1 } ;
    x2 = { 27 , 28 , 27 , 28 , 28 } ;
    x2 = x2 - 27 ;
    z  = { 1 0 0 0 0 ,                          /* design for cows  5 columns*/
           0 1 0 0 0 ,
           0 0 1 0 0 ,
           0 0 0 1 0 ,
           0 0 0 0 1 } ;
    ia = i(5) ;                                 /* relationship is diagonal */
    x1x1 = x1`*x1 ;  x1x2 = x1`*x2 ;  x1z = x1`*z ;
    x2x1 =x1x2` ;    x2x2 = x2`*x2 ;  x2z = x2`*z ;
    zx1 = x1z` ;     zx2 = x2z` ;     zzia = z`*z + lamb*ia  ;
    r1 = x1x1||x1x2||x1z ;                      /* left-hand side */
    r2 = x2x1||x2x2||x2z ;
    r3 = zx1  ||zx2  ||zzia ;
    lhs = r1//r2//r3 ;
    x1y = x1`*y ;                               /* right-hand side */
    x2y = x2`*y ;
    zy   = z`*y ;
    rhs = x1y//x2y//zy ;
    c = inv(lhs) ;
    b = c*rhs ;
    herd = b[1:2,] ; age = b[3,] ; cow = b[4:8,] ;      /*partition of results */
    print herd age cow ;  quit;
```

Example 6. ST animal model, related animals

Like (Ex.5), but cow 1 and cow 5 have the same sire (animal no. 6) and cow 2 and cow 4 have also the same sire (animal no. 7). Matrix Z will have now 7 columns. Which cow and which sire is the best? Compare results of example 5 and 6.

```
proc IML ; reset print ;
     h2 = 0.30 ;
     lamb = (1-h2)/h2 ;
     y = { 7000 , 8000 , 6000 , 9000 , 8000 } ;
     x1 = { 1  0 ,
            1  0 ,
            1  0 ,
            0  1 ,
            0  1 } ;
     x2 = { 27 , 28 , 27 , 28 , 28 } ;
     x2 = x2 - 27 ;
     z  = { 1 0 0 0 0 0 0 ,                          /* design for animals 7 columns*/
            0 1 0 0 0 0 0 ,
            0 0 1 0 0 0 0 ,
            0 0 0 1 0 0 0 ,
            0 0 0 0 1 0 0 } ;
     a = i(7) ; a[1,6] = 0.5 ; a[5,6] = 0.5 ; a[1,5] = 0.25 ;   /* animals relationship */
                a[6,1] = 0.5 ; a[6,5] = 0.5 ; a[5,1] = 0.25 ;
                a[2,7] = 0.5 ; a[4,7] = 0.5 ; a[2,4] = 0.25 ;
                a[7,2] = 0.5 ; a[7,4] = 0.5 ; a[4,2] = 0.25 ;
     ia = inv(a) ;
     x1x1 = x1`*x1 ;  x1x2 = x1`*x2 ;  x1z = x1`*z ;
     x2x1 = x1x2` ;   x2x2 = x2`*x2 ;  x2z = x2`*z ;
     zx1 = x1z` ;       zx2 = x2z` ;       zzia = z`*z + lamb*ia  ;
     r1 = x1x1||x1x2||x1z ;                          /* left-hand side */
     r2 = x2x1||x2x2||x2z ;
     r3 = zx1  ||zx2  ||zzia ;
     lhs = r1//r2//r3 ;
     x1y = x1`*y ;                                   /* right-hand side */
     x2y = x2`*y ;
     zy   = z`*y ;
     rhs = x1y//x2y//zy ;
     c = inv(lhs) ;
     b = c*rhs ;
     herd = b[1:2,] ; age = b[3,] ; cow = b[4:8,] ; sire = b[9:10,] ;
     print herd age cow sire ;
quit;
```

Example 7. BLUP from external files

Calculation with external files. Model like in (Ex.6). Total of 80 animals. Cows (animals 11-20, 22-50 and 61-80) in 8 herds, progenies of 11 sires (animals 1-10, 21). Animals are differently related. Older cows are mothers of younger ones. Animals 51-60 are young animals without production records and without progeny, connected by relationship with others animals. Identification numbers of animals correspond to generations, the oldest animal has smallest number. Missing parent is in pedigree file marked as 0. Levels of all effects are consecutively renumbered starting with 1. The command “array” is used for creating of design matrices and commands in cycles “do” is used for constructing of the relationship matrix according to Quaas (1976). External files are located in directory c:\ LinMod \myprog.

```
/*  .............          blupext          ..................*/
/* ..........   milk = HYS + age + animal  + e  ..........*/
```

```
filename   prod    "c:\LinMod\myprog\uzit" ;      /* production input file */
filename   ped     "c:\LinMod\myprog\rod" ;       /* pedigree input file */
filename   ebvs    "c:\LinMod\myprog\ebvcow" ;    /* output file of EBV */
data prod;                          /* prod = production*/
     infile prod ;
     input milk animal herd age do ;
proc means ;
proc freq ; tables herd ;
data y ;       /*..... creation files for matrices*/ /* y = milk */
     set prod ;
     keep milk ;
proc means ;
data x1 ;                           /* x1 = herd */
     set prod ;
     keep h1 - h8;                  /* according to number of herds */
     array x1 h1 - h8;
     do i = 1 to 8;                 /* set 0 to all elements of X1 */
     x1[i] = 0 ;
     end;
     do i = 1 to 8 ;                /* put 1 into position of observation in a herd */
     if herd = i then x1[i] = 1 ;
     end;
proc means;
data x2 ;                           /* x2 = age */
     set prod ;
     keep age ;                     /* one covariable */
     age = age -27 ;
proc means ;
data z ;                            /* z = animal */
     set prod ;
     keep j1 - j80 ;
     array z j1 - j80;              /* according to total number of animals including parents*/
     do i = 1 to 80;
        z[i] = 0 ;
     end;
     do i = 1 to 80 ;
     if animal = i then z[i] = 1 ;
        end;
proc means;
data pedig;                         /* pedig = pedigree*/
     infile ped;
     input anim sir mat ;           /* 0 = missing parent .....*/
proc means; run;
/*.....creation of relationship matrix A by Quass (1976), pedigree must be reordered
                                    ascending from the oldest animals.....*/
proc iml;
     use pedig;
     read all into b;               /* reading pedigree into matrix B with three columns */
     close pedig;
     n = nrow(b);                   /* no. of animals in pedigree */
     L=i(n);                        /* identity matrix */
     do i=1 to n;                   /* diagonal element of animal 1 */
```

```
        o = B[i,2]; m = B[i,3];
        if o = 0 & m = 0 then L[i,i] = 1;
        if o > 0 & m > 0 then do;
            x = L[o,1:o]; x = x#x;
            a = (sum(x))*0.25;
            y = L[m,1:m]; y = y#y;
            c = (sum(y))*0.25;
            L[i,i] = sqrt((1 - a - c));
        end;
        else if o > 0 then do;
            x = L[o,1:o]; x = x#x;
            a = (sum(x))*0.25;
            L[i,i] = sqrt((1-a));
        end;
        else if m > 0 then do;
            y = L[m,1:m]; y = y#y;
            c = (sum(y))*0.25;
            L[i,i] = sqrt((1-c));
        end;
/* continue in a given column with animal 2 and creation of overdiagonal element L[j,i];*/
        do j=i+1 to n;
            o = B[j,2]; m = B[j,3];
            if o = 0 & m = 0 then L[j,i] = 0;
            if o > 0 & m > 0 then L[j,i] = 0.5*(L[o,i] + L[m,i]);
            else if o > 0 then L[j,i] = 0.5*(L[o,i]);
            else if m > 0 then L[j,i] = 0.5*(L[m,i]);
        end;
    end;
    A = L* L`;                  /* relationship matrix A */
        /*..................................... BLUP equations  .....reading files into matrices.....*/
    h2 = 0.30 ;
    lamb = (1-h2)/h2 ;
    use y ;
    read all into y ;           /* reading file Y into matrix Y */
    close y ;
    use x1 ;
    read all into x1 ;              /* reading into X1 */
    close x1 ;
    use x2 ;
    read all into x2 ;              /* reading into X2 */
    close x2 ;
    use z ;
    read all into z ;               /* reading into Z */
    close z ;
    ia = inv(a) ;                                       /* construction of blocks for LHS */
    x1x1 = x1`*x1 ;   x1x2 = x1`*x2 ;   x1z  = x1`*z ;
    x2x1 = x1x2` ;    x2x2 = x2`*x2 ;   x2z  = x2`*z ;
    zx1   = x1z` ;      zx2   = x2z` ;      zzia = z`*z + lamb*ia  ;
    r1 = x1x1||x1x2||x1z ;      /* left-hand side */
    r2 = x2x1||x2x2||x2z ;
    r3 = zx1  ||zx2  ||zzia ;
    lhs = r1//r2//r3 ;
```

```
      x1y = x1`*y ;                 /* right-hand side */
      x2y = x2`*y ;
      zy   = z`*y ;
      rhs = x1y//x2y//zy ;
      c = inv(lhs) ;
      b = c*rhs ; print b ;
      herd = b[1:8,] ; age = b[9,] ;   animal = b[10:89,] ;
      print herd age animal ;
      create BVanim from animal ;/* file of EBV from vector of EBV of animals */
      append from animal ;
    /*..................... put breeding values with animal identification into file ..... */
data b ;
      set bvanim ;
      EBV = col1 ;         drop col1 ;
      animal = _n_  ;  /*creation of animal no. identification according to row no. in datafile */
proc sort data = prod ; by animal ;
data c ;
      merge prod b ; by animal ;      /*connecting EBV with production file*/
      file ebvs ;                     /* writing the file of EBV to directory*/
      put animal milk EBV herd age  ;
proc means ;
proc sort ; by ebv ;                  /* rank of animals */
proc print ;
run ;
/*..................................... finish ..................................................... */
```

Example 8. BLUP with direct calculation of inversion of relationship

Like (Ex.7), with direct creation of $\mathbf{A}^{-1}$ according to Henderson (1976), this is usable for large data.

```
/*  .............              blupdir          ..................*/
/* ..........  milk = HYS + age + animal  + e  ..........*/
filename   prod    "c:\LinMod\myprog/uzit" ;      /* production input file */
filename   ped     "c:\LinMod\myprog/rod" ;       /* pedigree input file */
filename   ebvs    "c:\LinMod\myprog/ebvcow2" ;   /* output file of EBV */
data prod;                                                    /* prod =  production*/
      infile prod ;
      input milk animal herd age sp ;
proc means ;
proc freq ; tables herd ;
data y ;                                      /* y = milk */
      set prod ;
      keep milk ;
proc means ;
data x1 ;                                     /* x1 = herd */
      set prod ;
      keep h1 - h8;                           /* according to number of herds */
      array x1 h1 - h8;
      do i = 1 to 8;
            x1[i] = 0 ;
      end;
```

```
      do i = 1 to 8 ;
           if herd = i then x1[i] = 1 ;
      end;
proc means;
data x2 ;                                   /* x2 = age */
      set prod ;
      keep age ;                            /* one covariable */
      age = age -27 ;
proc means ;
data z ;                                    /* z = animal */
      set prod ;
      keep j1 - j80 ;
      array z j1 - j80;          /* according to total number of animals including parents*/
      do i = 1 to 80;
           z[i] = 0 ;
      end;
      do i = 1 to 80 ;
           if animal = i then z[i] = 1 ;
      end;
proc means;
data pedig;                                 /*pedig = pedigree*/
      infile ped;
      input anim sir mat ;                  /* 0 = missing parent */
proc means; run;
/*.....Direct creation of inverted relationship matrix inv(A) ..by Henderson (1976)..... */
proc iml;
      use pedig;
      read all into b;
      close pedig;
      n = nrow(b);                          /* animals in pedigree */
      ia=j(n,n,0);                          /* matrix with 0 */
      do i = 1 to n ;
           an = b[i,1] ; si = b[i,2]; ma = b[i,3];
           if si = 0 & ma = 0 then do;      /* both parents unknown*/
                ia[an,an] = ia[an,an] + 1;  /* adding value to the position in IA */
           end;
           else if si > 0 & ma = 0 then do; /* mother unknown*/
                ia[an,an] = ia[an,an] + (4/3) ;
                ia[an,si]  = ia[an,si] - (2/3) ;
                ia[si,an]  = ia[an,si] ;
                ia[si,si]   = ia[si,si] + (1/3) ;
           end;
           else if si = 0 & ma > 0 then do ;/* sire unknown*/
                ia[an,an]   = ia[an,an] + (4/3) ;
                ia[an,ma]  = ia[an,ma] - (2/3) ;
                ia[ma,an]  = ia[an,ma] ;
                ia[ma,ma] = ia[ma,ma] + (1/3) ;
           end;
           else if si > 0 & ma > 0 then do; /* both parents known*/
                ia[an,an]   = ia[an,an] + 2;
                ia[an,si]    = ia[an,si] - 1;
                ia[si,an]    = ia[an,si] ;
```

```
                ia[an,ma]  = ia[an,ma] - 1;
                ia[ma,an]  = ia[an,ma] ;
                ia[si,si]    = ia[si,si] + (1/2) ;
                ia[si,ma]   = ia[si,ma] + (1/2) ;
                ia[ma,si]   = ia[si,ma] ;
                ia[ma,ma] = ia[ma,ma] + (1/2);
            end;
        end;
/*.............................................. BLUP equations  .....reading files into matrices.....*/
        h2 = 0.30 ;
        lamb = (1-h2)/h2 ;
        use y ;
        read all into y ;                              /* reading into matrices */
        close y ;
        use x1 ;
        read all into x1 ;
        close x1 ;
        use x2 ;
        read all into x2 ;
        close x2 ;
        use z ;
        read all into z ;
        close z ;
        x1x1 = x1`*x1 ;  x1x2 = x1`*x2 ;  x1z = x1`*z ;
        x2x1 = x1x2` ;    x2x2 = x2`*x2 ;  x2z = x2`*z ;
        zx1 = x1z` ;       zx2 = x2z` ;       zzia = z`*z + lamb*ia  ;
        r1 = x1x1||x1x2||x1z ;                         /* left-hand side*/
        r2 = x2x1||x2x2||x2z ;
        r3 = zx1  ||zx2  ||zzia ;
        lhs = r1//r2//r3 ;
        x1y = x1`*y ;                                  /* right-hand side*/
        x2y = x2`*y ;
        zy   = z`*y ;
        rhs = x1y//x2y//zy ;
        c = inv(lhs) ;
        b = c*rhs ;          print b ;
        herd = b[1:8,] ; age = b[9,] ;   animal = b[10:89,] ;
        print herd age animal ;
        create BVanim from animal ;                    /* vector of BV of animals */
        append from animal ;
data b ;                    /* put breeding values with animal identification into file */
        set bvanim ;
        EBV = col1 ;          drop col1 ;
        animal = _n_  ;    /*creation of animal no. identification according to row no. in datafile*/
proc sort data = prod ; by animal ;
data c ;
        merge prod b ; by animal ;
        file ebvs ;                                    /* writing the file of EBV */
        put animal milk EBV herd age  ;
proc means ;
proc sort ; by ebv ;
proc print ;
```

```
run ;
/*..................................... finish ..................................... */
```

Regression coefficients of loci by RRBLUP and calculation of DGV.

Genetic chips with detection of huge number of genetic markers single nucleotide polymorphism (SNP) are used for genotyping of animals. Example of laboratory output is in an attached directory ./LinMod/multist/. Alphabetic laboratory results for alleles are converted into numerical form expressing the number of second allele in a locus. Values of all loci are analysed in a joint simultaneous analysis. Number of loci is usually bigger than number of genotyped animals in referenced input data therefore the special algorithms which allow solutions are used. One of the simplest ways is a mixed model RRBLUP, adding some values to diagonal and considering each locus as a random effect. Therefore the name of procedure is the Ridge Regression or Random Regression.

RRBLUP procedure of prediction of genomic enhanced breeding value (GEBV) is based on prediction of SNP regression coefficients of all loci according phenotypes of animals in a reference population. These regression coefficients are then used for prediction of direct genetic value (DGV) of young animals (Meuwissen et al., 2001; Szyda et al., 2011; Pešek et al. 2014). The assumption of the method is that genetic variability of all loci is similar.

Input data for a calculation are "pseudo-phenotypes" daughter yield deviations (DYDs) or their approximations deregressed proofs (DRPs) calculated backward from EBVs of reliably proven sires (Schaeffer 1994; Jairath et al. 1998). These values are free from influence of systematic environmental effects and contain only the genetic component of sire and random error. In a simple case, when EBV of sire is influenced mainly by progeny, and others sources of information are negligible, DRP can be approximated by dividing EBV by reliability (Rel). Reliabilities of input EBVs are used for calculations of effective daughter contributions (EDCs), which are used as weights in a weighted analysis.

$$EDC = k*(Rel)/(1-Rel) \quad , \qquad (14)$$

where: EDC is effective daughter contribution,
Rel is reliability of sire`s EBV,
k is the ratio of variances adequate to progeny test

$$k = (4 - h^2)/ h^2 \qquad (15)$$

Regression coefficients for loci is then calculate according to model equation

$$DRP = X1b + T1v + e \quad , \qquad (16)$$

where **DRP** is known vector of input pseudo-phenotype data DRPs, with weights EDCs located in diagonal matrix **W**,
X1 is a matrix assigning DRPs of proven bulls to fixed effects,
b is an estimated fixed effect (usually one common constant),
T1 is a matrix assigning DRPs of proven bulls to regression coefficients of each locus, with values at each locus <0, 1, 2> according to number of second allele,
v is a vector of predicted random effects – SNP regression coefficients, and
e is random error.

System of normal equation for prediction of SNP regression coefficients is as follows:

$$\begin{bmatrix} X1'WX1 & X1'WT1 \\ T1'WX1 & T1'WT1 + \lambda I \end{bmatrix} \begin{bmatrix} b \\ v \end{bmatrix} = \begin{bmatrix} X1'WDRP \\ T1'WDRP \end{bmatrix} \qquad (17)$$

where **W** is a diagonal matrix of weights containing EDCs of proven bulls on the diagonal,
I is an identity matrix of the size according the number of loci (m),

λ is the variance ratio of residual variability divided by average genetic variance of one locus of all treated SNPs loci, which is equal to

$$\lambda = m*k \quad , \quad (18)$$

Prediction of DGV of young unproven animals is:

$$\mathbf{DGV} = \mathbf{X2\hat{b}} + \mathbf{T2v} \quad , \quad (19)$$

where **X2** is a matrix assigning DGVs of young animals to fixed effects (one common constant),

T2 is a matrix assigning DGVs of young animals to regression coefficients

For prediction of GEBV, DGV is combined with pedigree based EBV using selection index

$$GEBV = \alpha1*DGV + \alpha2*PA \quad , \quad (20)$$

where α1, α2 are weights in a selection index,

PA is a pedigree based EBV of young animal (parent average).

Example 9. Regression coefficients of loci

Using attached files predict SNP regression coefficients in population of proven bulls and use this regression coefficient for prediction of DGV and GEBV of young unproven bulls. Only small number of genetic loci is in example (15 loci), from which according the quality checking for MAF are 2 eliminated, therefore only 13 loci are used for prediction. In a reference population are 10 genotyped proven sires (animals 1 - 10) with sufficient reliability of EBV. 4 genotyped young bulls have only pedigree based EBV with low reliability (animals 11 - 14). In a practical case the size of data would be much bigger therefore the solution of the system of equation (17) is not by inversion of LHS, but by iterative procedure. Efficient algorithms, as Preconditioned Conjugate Gradient (PCG) (Lidauer et al., 1999; Tsuruta et al., 2001; Legara, Misztal, 2008), are used for solution. Here we use for simplicity technique based on Gauss-Seidel (GS) iteration.

Files are located in ./LinMod/multist/rrblu/.

```
/*..................................... regreSNP.sas.....................................*/
/*.................................Petr Pesek..... Genetic Days 2014.....................*/
/*.....................Construction of matrix of genetic markers.....................*/
/*Estimation of regression coefficients by ridge regression method RRBLUP*/
/*.........................Calculation of direct genetic value DGV.....................*/
Filename Genot "C:/ LinMod /multist/rrblu/Gen.txt";          /*input genot*/
Filename EBV   "C:/ LinMod /multist/rrblu/EBV.txt";    /*input EBV*/
Filename HELP "C:/ LinMod /multist/rrblu/HELP.txt";  /*Help file*/
Filename pred   "C:/ LinMod /multist/rrblu/pred";         /*output DGV*/
Filename matT  "C:/ LinMod /multist/rrblu/matt";         /*output matr T*/
                    /*.................................input files.....................*/
data DYD;
     infile EBV; input animal EBV rel; /*importing animal + EBV + Rel*/
     DYD=EBV/rel;                      /*calculating daughter yield deviations*/
     h2=0.3;                           /*heritability*/
     k=(4-h2)/h2;                      /*variance ratio*/
     EDC=k*rel/(1-rel);                /*effective daughter contribution*/
     drop  rel h2 k;
proc sort; by animal;
proc means ;
data SNP;
     infile Genot;  input animal SNP genotype;
proc sort; by animal;
```

```
proc means ;
data ALL;
      merge DYD SNP; by animal;
      drop DYD EDC;
      proc sort; by SNP;
data _null_;                  /*creating empty data*/
      keep animal SNP genotype;
      file help;
      set ALL; by SNP;
        /*.....initial genotype sum and number of bulls with known genotype in the SNP.....*/
      if first.snp then do;
            sum=0; numb=0;
      end;
                                    /*if genotype in the SNP is not missing then do*/
      if genotype ne . then do;
            numb+1;           /*add one to number of bulls with known genotype*/
            sum+genotype;     /*add genotype value (0,1,2) to total genotype sum in the SNP*/
      end;
/*if last number of SNP, then put SNP number and sum of genotypes into output help file*/
      if last.snp then put SNP numb sum;
data MAF1;
      infile help; input SNP numb sum;
      meangenot=sum/numb;/*calculating mean genotype in the SNP*/
      if meangenot<0.1 or meangenot>1.9 then delete;
proc sort; by SNP;
data Maf2;
      set MAF1;
      nSNP=_n_;               /*renumbering loci*/
Data edSNP;
      merge ALL MAF2; by SNP;
      if meangenot="." then delete;
      keep animal nSNP genotype;
proc means ;
                  /*.................................files into matrices .....................................*/
proc iml;
      start;
      use DYD; Read all into DYD;       /*read work DYD into matrix DYD*/
      use edSNP; Read all into SNP;     /*read worked SNP into matrix SNP*/
      BULL=DYD[,1];                     /*bulls numbers*/
      nprBULL=10;                       /*number of proven bulls*/
      nyBULL=4;                         /*number of young bulls*/
      nBULL=nprBULL+nyBULL;             /*total number of bulls*/
      nSNP=max(SNP[,3]);                /*number of SNPs*/
      W=J(nprBULL,nprBULL,0);           /*creating diagonal matrix containing weights EDC*/
      do i=1 to nprBULL;
            W[i,i]=DYD[i,4];
      end;
      DYDpr=DYD[1:nprBULL,3];           /*reading block of DYD only proven bulls*/
      X1=J(nprBULL,1,1);                /*vector of ones for proven bulls*/
      X2=J(nyBULL,1,1);                  /*vector of ones for young bulls*/
      T=J(nBULL,nSNP,.);                /*creating free matrix T for all bulls*/
      nrow=nSNP*nBULL;                  /*number of rows in matrix SNP*/
```

```
do i=1 to nrow;                    /*number of iteration according rows in SNP matrix*/
      BULL=SNP[i,1];               /*reading bull number*/
      locus=SNP[i,3];              /*reading locus number*/
      genot=SNP[i,2];              /*reading genotype*/
      T[BULL,locus]=genot;         /*writing locus genotype of the bull into T*/
end;
T1=T[1:nprBULL,];                  /*cutting block for proven bulls*/
T2=T[(nprBULL+1):nBULL,];          /*cutting block for young bulls*/
h2=0.3;
lamb=(4-h2)/h2;
f=lamb*nSNP;
I=i(nSNP);
          /*.................creating system of normal equation for proven bulls only.....*/
XWX=X1`*W*X1;   XWT=X1`*W*T1;
TWX= XWT `;         TWT=T1`*W*T1 + I*f;
LHS1=XWX||XWT;                     /* left hand side */
LHS2=TWX||TWT;
LHS=LHS1//LHS2;
RHS1=X1`*W*DYDpr;                  /* right hand side */
RHS2=T1`*W*DYDpr;
RHS=RHS1//RHS2;
           /*.....................................iterative solution.....................................*/
b=j(nSNP+1,1,0);                   /*initial vector of solutions with 0 */
b0 = b ;                           /* storing of initial step */
numit=nSNP+1;                      /*number of iterations according to
                                    number of SNPs + common constant*/
do j=1 to 50000;                   /* number of maximal repetitions of iterations*/
     do i=1 to numit;
          RHS1=LHS*B;              /*calculating RHS using vector of solutions*/
          D=RHS[i]-RHS1[i];        /*difference between real and calculated RHS*/
          R=D/LHS[i,i];            /*dividing difference by the diagonal element*/
          B[i]=B[i]+R;             /* update vector of solution */
        end;
     D = b0 - b ;                  /*difference vector previous and current solution */
     D=abs(D);
     DIFF=max(D);                  /*largest abs. differ. of previous and current solution*/
     if DIFF<10e-8 then goto fin;  /*skip to fin if absolute value difference is smaller
                                    than 10e-8*/
     b0 = b ;
end;
fin: print j diff ;                /* print round of termination */
           /*.................predicting direct genetic values for young bulls.............*/
DGV=(x2||T2)*b;
EBV=DYD[(nprBULL+1):nBULL,2];  /* input pedigree of young bulls */
GEBV=EBV*0.2+DGV*0.8;              /*predicting genomic breeding values*/
print b DGV GEBV;
predic = DGV||GEBV ;
create pred from predic ;          /* prediction of BV into file */
append from predic ;
create matT from T ;               /* matrix of genotypes into file */
append from T ;
finish;
```

```
run; quit;
            /*.....writing files to directory.....*/
data pred ;                          /* storing BV */
    set pred ;
    anim = _n_ + 10 ;                /*identification of animal - rank of young bulls*/
    file pred ; put anim col1 col2 ;
proc means ;
data matT ;                          /* storing matrix T */
    set matT ;
    anim = _n_ ;                     /*identification of animal - rank of bulls */
    file matT ; put anim col1 - col13 ;
proc means ;
run;
          /*..... finish .....*/
```

GBLUP

Regression coefficients **v**s are used for prediction DGV. DGV should be alike the EBV predicted by common procedure of BLUP - animal model. Variances/covariances of EBVs between animals are $\mathbf{A}*\sigma^2_u$. Variances/covariances of DGVs between animals are following (19) $\mathbf{T*v*v'*T'} = \mathbf{T*T'*}\ \sigma^2_v$, where σ^2_v is the genetic variance of loci with regression coefficients. Expectations of variances of EBVs and DGVs should be similar:

$$\mathbf{A}*\sigma^2_u \sim \mathbf{T*T'*}\ \sigma^2_v$$

from which arise

$$\mathbf{A} \sim \mathbf{T*T'*}\ \sigma^2_v / \sigma^2_u \qquad (21)$$

T*T' is the bases for *realised genomic relationship matrix* **G** between animals calculated according similarity of segments of genom. The scale of **A** and **G** should be similar and both should express the relationship of animals with respect to the unselected ancestors in a base population. Alleles of animals in a base population are usually not known therefore alleles in a current population of living animals are used. **G** is then standardised (regressed) according **A**. Methodology of calculation of **G** follows for example from VanRaden P. M. (2008); Forni et al. (2011) and Vitezica et al. (2011):

$$G = \frac{(T-Q)(T-Q)'}{trace((T-Q)(T-Q)')/n} \qquad (22)$$

where **G** is the realised genomic relationship matrix,
T is matrix of SNP genetic markers wit values <0, 1, 2>,
Q is matrix with columns of averages from **T** (average allele frequencies in loci),
n is the number of genotyped animals.

Values of **G** are shifted, so that the elements of the pedigree relationship matrix only for genotyped animals $\mathbf{A_{22}}$ and elements of **G** would have the same averages.

GBLUP (VanRaden, 2008) is based on substitution of matrix **G** instead **A** into linear model for calculation of *DGV*:

$$DRP = Xb + Zu + e \quad , \qquad (23)$$

where **DRP** is known vector of input pseudo-phenotype data DRPs, with weights EDCs located in diagonal matrix **W**,
Xb covers usually only one common constant in a model,
u is unknown vector of predictions of DGVs.

The system of normal equations is modified into a form of sire-model:

$$\begin{bmatrix} X'WX & X'WZ \\ Z'WX & Z'WZ + kG^{-1} \end{bmatrix} \cdot \begin{bmatrix} b \\ u \end{bmatrix} = \begin{bmatrix} X'WDRP \\ Z'WDRP \end{bmatrix} \qquad (24)$$

where $k = (4 - h^2) / h^2$

Example 10. Genomic relationship in BLUP

Like (Ex.9), evaluated by GBLUP method. Matrix of genetic markers **T** contains both parts **T1** and **T2** for proven animals with known DRPs and young animals without production records. The size of system of equations agrees with number of genotyped animals (n) + 1 for common constant. Files are located ./LinMod/multist/gblu/.

```
                /*.....................................GBLUP.....................................*/
                /*...........Calculation of direct genetic value DGV.....................*/
Filename EBV   "c:/LinMod/multist/gblu/EBV.txt";        /*input EBV*/
Filename matT  "C:/ LinMod /multist/gblu/matt";         /*input matrix T*/
Filename predg  "C:/ LinMod /multist/gblu/predg";       /*output DGV*/
          /*.....................................................input files.....................................................*/
data prod;
     infile EBV; input animal EBV rel; /*importing EBV + Rel*/
    if animal > 10 then delete ;          /* use only proven sires*/
    DYD=EBV/rel;                          /*calculating daughter yield deviations*/
    h2=0.3;                               /*heritability*/
    k=(4-h2)/h2;                          /*variance ratio*/
    EDC=k*rel/(1-rel);                    /*effective daughter contribution*/
proc means ;
data drp ;                                /* pseudo-phenotype records */
    set prod ;
    keep dyd ;
data weig;                                /* weights */
    set prod ;
    keep edc ;
data x ;                                  /* x = common constant */
    set prod ;
    keep h;                               /* according to number of herds */
    h = 1 ;
data z ;                                  /* z = animal*/
    set prod ;
    keep j1 - j14 ;
    array z j1 - j14;                     /* according to total number of evaluated animals */
    do i = 1 to 14;                       /* file of "0" */
        z[i] = 0 ;
    end;
    do i = 1 to 14 ;              /* design matrix for animals „1“ to position with production*/
        if animal = i then z[i] = 1 ;
    end;
proc means;
data matT ;                               /* reading matrix T */
    infile matT ;
    input animal loc1 - loc13 ;
proc means ;
     /*.....................................G ...genomic relationship .....................................*/
```

```
proc iml ;
      use matT;
      read all into gt;
      close matt;
      t = gt[,2:14];       print t;              /* 13 loci */
      nsn = ncol(t) ;                            /* number of SNPs for animal */
      ng  = nrow(t) ;                            /* number of genotyped animals */
      ones = j(1,ng,1);
      suones = ones * t;
      aver = suones /(ng);                       /*vector of averages of second allele */
      q = j(ng,nsn,1);                           /* matrix of averages Q */
      q = aver # q;
      print q ;
      tq = t - q;
      g = tq*tq`;                                /* numerator in (22) */
      deno = trace(g)/ng;                        /* denominator in (22) */
      gg = g/deno ;                              /* matrix G */
      print gg ;
      gg = 0.99*gg + 0.01*(i(ng)) ;              /*warrant inversion*/
      ig = inv(gg) ;                             /* inversion of G */
          /*..................................................... BLUP equations .....................................................*/
      h2 = 0.30 ;
      lamb = (4-h2)/h2 ;
      use drp ;
      read all into y ;                          /* reading DRP into matrices */
      close drp ;
      use weig ;
      read all into we ;                         /* reading weights */
      close weig ;
      w = diag(we) ;
      use x ;                                    /* reading X */
      read all into x ;
      close x ;
      use z ;                                    /* reading Z */
      read all into z ;
      close z ;
      xx = x`*w*x ;  xz    = x`*w*z ;
      zx = xz` ;        zzig = z`*w*z + lamb*ig  ;
      r1 = xx||xz ;                              /* system of equations */
      r2 = zx||zzig ;
      lhs = r1//r2 ;
      xy = x`*w*y ;
      zy   = z`*w*y ;
      rhs = xy//zy ;
      c = inv(lhs) ;
      b = c*rhs ;
      constant = b[1,] ;   animal = b[2:15,] ;
      print constant animal ;
      create BVanim from animal ;                /* vector of BV of animals */
      append from animal ;
          /*..................................................... output files .....................................................*/
data b ;
```

```
    set bvanim ;
    DGV = col1 ;        drop col1 ;
    animal = _n_  ;                    /* identific. no. of animals */
data c ;
    merge prod b ; by animal ;         /* merging with input production records */
    file predg ;                       /* writing the file of EBV */
    put animal 1-2 ebv 4-8 dyd 10-14 rel 16-20 2 DGV  22- 29 2;
proc means ;
proc sort ; by DGV ;
proc print ;
run ; /*..................................................... finish ..................................................... */
```

ssGBLUP

Misztal I. et al. (2009) and Christensen and Lund (2010) developed a single-step procedure *ssGBLUP*, which overcomes several critical assumptions required by multi-step procedures. The procedure combines nation-wide files of production records and pedigree with genomic information and allows common rank of all genotyped and un-genotyped animals. Calculation produces directly *GEBV* exploiting all information. Přibyl et al. (2012); (2013) and (2014) used this methodology for the genetic evaluation of the Czech Holstein population and for combination in one common evaluation nation-wide databases with all available Interbull DRPs.

ssGBLUP is an extension of common BLUP procedure according (10) and (12) by augmenting the pedigree relationship matrix **A** into **H**. In a system of BLUP equations is used the inverse of relationship matrix. The inverse of **H** is:

$$H^{-1} = A^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & F \end{bmatrix} \quad , \qquad (25)$$

where **F** corresponds with a segment of relationship matrix for genotyped animals

$$\mathbf{F} = \omega\,(\mathbf{G}^{-1} - \mathbf{A}^{-1}_{22}) \quad , \qquad (26)$$

where ω is the weight < 0 , 1 > of genomic relationship. It is expected that genetic markers not explain entire genetic variability, value around ω ~ 0.8 (80 %) is used therefore,

G is a genomic relationship matrix,

$\mathbf{A}_{22}$ is a part of pedigree relationship matrix corresponding only with genotyped animals. $\mathbf{A}^{-1}_{22}$ is subtracted from $\mathbf{H}^{-1}$ to prevent the double counting of relationship.

Example 11. Single step GEBV

Like (Ex.7). But animals 2, 5, 21 and 51-60 are genotyped. From these 2, 5 and 21 are progeny tested sires, animals 51-60 are young. As an example only 40 SNP loci are used. Files are stored in ./LinMod/myprog/.

```
                /* ..................................... ssgblup .....................................*/
                /*          switching type of G matrix .....................row 148          */
                /* ..................... milk = HYS + age + animal  + e  .....................*/
filename  prod   "c:\LinMod\myprog/uzit" ;        /* production input file */
filename  ped    "c:\LinMod\myprog/rod" ;         /* pedigree input file */
filename  gen    "c:\LinMod\myprog/genot" ;       /* SNP genotype input file */
filename  genan  "c:\LinMod\myprog/sezgenot" ;    /* list genot input file */
filename  gebvs  "c:\LinMod\myprog/gebv" ;        /* output file of GEBV */
```

```
filename   ebvs     "c:\LinMod\myprog/ebvcow" ;   /* input file of previous EBV */
filename   gemat   "c:\LinMod\myprog/gemat" ;      /* output G matrix triangle */
              /*..................................................... production.....................................*/
data prod;    title " production file " ;
      infile prod ;
      input milk animal herd age dopen ;
proc means ;
proc freq ; tables herd ;
data y ;                                    /* y = milk */
      set prod ;                  title " vector Y " ;
      keep milk ;
proc means ;
data x1 ;                                   /* X1 = herd */
      set prod ;
      keep h1 - h8;                         /* according to number of herds */
      array x1 h1 - h8;
      do i = 1 to 8;                        /* set 0 to all elements of X1 */
            x1[i] = 0 ;
      end;       title " matrix X1 " ;
      do i = 1 to 8 ;                 /* put 1 into position of observation in a herd */
            if herd = i then x1[i] = 1 ;
      end;
proc means;
data x2 ;                                   /* X2 = age */
      set prod ;      title " matrix X2 " ;
      keep age ;                        /* one covariable */
      age = age -27 ;
proc means ;
data z ;                                    /* Z = animal */
      set prod ;
      keep j1 - j80 ;
      array z j1 - j80;                 /* according to total number of animals including parents*/
      do i = 1 to 80;
            z[i] = 0 ;
      end;       title " matrix Z " ;
      do i = 1 to 80 ;
            if animal = i then z[i] = 1 ;
      end;
proc means;
data genot ;                              /* genotypes SNP */
      infile gen ;
      input gan g1 - g40 ;    title " genotypes " ;
proc means ;
data listg ;                              /* list of genotyped animals */
      infile genan ;
      input gan ;    title " list of G animals " ;
proc means ;
            /*.....................................................pedigree.....................................................*/
data pedig;                     title " pedigree " ;
      infile ped;
      input anim sir mat ;          /* 0 = missing parent */
proc means; run;
```

```
        /*.....relationship    A    .....*/
proc iml;
use pedig;
    read all into b;
    close pedig;
    n = nrow(b);                /* animals in pedigree */
    L=i(n);                     /* unity matrix */
    do i=1 to n;                /* diagonal element of animal 1 */
        o = B[i,2]; m = B[i,3];
        if o = 0 & m = 0 then L[i,i] = 1;
        if o > 0 & m > 0 then do;
            x = L[o,1:o]; x = x#x;
            a = (sum(x))*0.25;
            y = L[m,1:m]; y = y#y;
            c = (sum(y))*0.25;
            L[i,i] = sqrt((1 - a - c));
        end;
        else if o > 0 then do;
            x = L[o,1:o]; x = x#x;
            a = (sum(x))*0.25;
            L[i,i] = sqrt((1-a));
        end;
        else if m > 0 then do;
            y = L[m,1:m]; y = y#y;
            c = (sum(y))*0.25;
            L[i,i] = sqrt((1-c));
        end;
  /*..... continue in a given column with animal 2 and creation of overdiagonal element L[j,i];*/
        do j=i+1 to n;
            o = B[j,2]; m = B[j,3];
            if o = 0 & m = 0 then L[j,i] = 0;
            if o > 0 & m > 0 then L[j,i] = 0.5*(L[o,i] + L[m,i]);
            else if o > 0 then L[j,i] = 0.5*(L[o,i]);
            else if m > 0 then L[j,i] = 0.5*(L[m,i]);
        end;
    end;
    A = L* L`;                  /* relationship matrix A */
        /*..... A22.....of genotyped animals .....*/
    use listg;
    read all into lg;
    close listg;
    ng = nrow(lg);              /* number of genotyped animals */
    a22 = j(ng,ng,0);
    do i = 1 to ng;
        f = lg[i];
        do j = 1 to ng;
            d = lg[j];
            a22[i,j] = a[f,d];  /* from A into A22 */
        end;
    end;
    print a22;
        /*.....G ...genomic relationship .....*/
```

```
use genot;
read all into gt;
close genot;
t = gt[,2:41];
nsn = ncol(t) ;                 /* number of SNPs for animal */
ones = j(1,ng,1);
suones = ones * t;
aver = suones /(ng);            /*vector of averages of second allele */
q = j(ng,nsn,1);                /* matrix of averages Q */
q = aver # q;
print q ;
tq =t  - q;
print tq ;
g = tq*tq`;                     /* numerator of (22) */
deno = trace(g)/ng;             /* denominator of (22) */
gg = g/deno ;
      /* ..... triangle G into file .....*/
velslg = (ng*ng - ng)/2 + ng;          /* size of file for triangle*/
slog = j(velslg,3,0);
k = 1;
do i = 1 to ng;
      do j = 1 to i;
            slog[k,1] = lg[i];
            slog[k,2] = lg[j];
            slog[k,3] = gg[i,j];
            k = k + 1 ;
      end ;
end ;
create mage from slog;
append from slog;                  /* end of file */
ggc = gg - a22 ;                   /* scaling of G */
correct = (ones * ggc * ones`)/(ng*ng) ;
ggc = gg + correct ;
print gg; print correct ; print ggc ;
      /*.....alternative of G .....*/
*ggc = gg ;                        /* without correction for A22*/
      /*.....iv(H) ....inversion of combined relationship .....*/
omeg = 0.8 ;
ggg = 0.99*ggc + 0.01*a22;         /* warrant the inversion */
gin = inv(ggg);                    /* inversion G */
a22in = inv(a22);                  /* inversion A22 */
f = omeg*(gin - a22in);
co = j(n,n,0);                     /* extension of F over matrix of all animals */
do i = 1 to ng;                    /* ng = number of genotyped animals */
      e= lg[i];                    /* lg = list of genotyped animals */
      do j = 1 to ng;
            d = lg[j];
            co[e,d] = f[i,j];
      end ;
end ;
ia = inv(a);
ih =ia + co ;                      /* inversion H */
```

```
        /*.....BLUP equations .....*/
    h2 = 0.30 ;
    lamb = (1-h2)/h2 ;
    use y ;
    read all into y ;                    /* reading file Y into matrix Y */
    close y ;
    use x1 ;
    read all into x1 ;
    close x1 ;
    use x2 ;
    read all into x2 ;
    close x2 ;
    use z ;
    read all into z ;
    close z ;
    x1x1 = x1`*x1 ;  x1x2 = x1`*x2 ;  x1z = x1`*z ;
    x2x1 = x1x2` ;   x2x2 = x2`*x2 ;  x2z = x2`*z ;
    zx1 = x1z` ;      zx2 = x2z` ;       zzia = z`*z + lamb*ih  ;     /*inclusion of inv(H) */
    r1 = x1x1||x1x2||x1z ;       /* left-hand side*/
    r2 = x2x1||x2x2||x2z ;
    r3 = zx1  ||zx2  ||zzia ;
    lhs = r1//r2//r3 ;
    x1y = x1`*y ;                /* right-hand side*/
    x2y = x2`*y ;
    zy   = z`*y ;
    rhs = x1y//x2y//zy ;
    c = inv(lhs) ;
    b = c*rhs ;
    herd = b[1:8,] ; age = b[9,] ;   animal = b[10:89,] ;
    print herd age animal ;
    create BVanim from animal ;  /* file of GEBV from vector of GEBV of animals */
    append from animal ;
          /*..... file .....*/
data gm ;                        /* writing G into file */
    set mage ;     title " G matrix " ;
    file gemat ;
    put col1 1-10 col2 11-20 col3 21-30  5 ;
proc means ;
data b ;                         /*  GEBV of animals */
    set bvanim ;
    GEBV = col1 ;       drop col1 ;
    animal = _n_  ;    /*creation of animal no. identification according to row no. in datafile*/
proc sort data = prod ; by animal ;
data c ;
    merge prod b ; by animal ;/*connecting EBV with production */
    file gebvs ;                 /* writing the file of GEBV */
    put animal milk GEBV herd age  ;
data d ;
    infile ebvs ;                /* input of previous BLUP */
    input animal milk EBV herd age  ;
    keep animal ebv ;
data e ;                    title " GEBV and EBV " ;
```

```
    merge c d ; by animal ;     /* compare EBV and GEBV */
proc sort ; by ebv ;            /* rank of animals */
proc corr ; var gebv ebv ;
data young ;                    /* young animals */
    set e ;
    if animal < 51 then delete ;
    if animal > 60 then delete ;
proc corr ; var gebv ebv ;
proc print data=e;
run ; /*..... finish ..... */
```

5. BLUPF90-family programs

BLUPF90-family of programs (Misztal et al., 2002) is a collection of software for variance components and mixed models calculation covering several methodologies for linear and threshold traits. Different programs run independently. BLUPF90 programs are available for Linux, Windows and Mac. Executable files could be copied into computer of user without special installation. The programs are free for research but their use should be acknowledged in publications. For commercial use please contact Ignacy Misztal. Basic manuals are *remlf90.pdf* and *blupf90.pdf*, at the end of parameter file is possible append "options". Detailed informations are available on http://nce.ads.uga.edu/~ignacy/ and http://nce.ads.uga.edu/wiki/doku.php.

User can participate in a discussion group by registering on this wiki page https://groups.yahoo.com/neo/groups/blupf90/info.

Three basic files are used for calculation - *data*, *pedigree* and *parameter file*. Levels of all effects in a data files must by renumbered from 1. Items in a data file are in a free format separated by space. There are three mandatory columns in a pedigree file - animal, parent1, parent2 and eventually coefficient (parent code). It is also possible to define phantom parents groups in pedigree file, which are coded with numbers highest than numbers of animals. If no specified, missing values are "0". Output log from program run is displayed on screen (or can be redirected into file), the solutions are saved in file *solutions*. There are four basic columns in the file solutions: identification of trait, effect, level of effect and solution.

All mandatory checks of data and pedigree files together with parameterizations can be done by the program *renumf90*. The program *pregsf90* can be used to handling files of genetic markers and preparing genomic relationship. For more detailed info visit http://nce.ads.uga.edu/wiki/

Running under Linux

Programs should be copied into directory */user/local/bin*. All your files (*data*, *pedigree* and *parameter file*) had to be in same directory. The appropriate program, for example *blupf90*, is starting from the command line by the command: *blupf90 > comment* , which redirects output from the screen into the *comment* file. After enter this command, the cursor is waiting on the next line. You have to write name of parameter file (e.g.. *param)* and enter again. If you start program which is located in other directory, the command had to lead with ./ (e.g. *./blupf90 > comment*, if executable programs are in your current directory).

Running under Windows

In a simple case the executable file of program, for example *blupf90.exe* could be copied into your directory with data, pedigree and parameter files. In the same directory locate the batch file, for example *blupf90.bat*. Execution is by submitting ".bat" file and then typing the name of parameter file *param*.

Example 12. Single-trait BLUP

Like (Ex.7), BLUP prediction of EBV. Files are stored in ./LinMod/blupf90/sintrait/. Prior known heritability of analysed trait is 0.30.

The code of batch file blupf90.bat (can be edited by whatever text editor):

```
echo off
echo type name of parameter file
blupf90.exe > comment          /*name of output commentary file
echo finish of calculation     from an execution of blupf90 program
```

Parameter file paramST:

```
#          paramST
# single trait BLUP animal model
# milk = HYS + age + animal + e
DATAFILE
uzit                          # name of input data file
NUMBER_OF_TRAITS
1
NUMBER_OF_EFFECTS
3
OBSERVATION(S)
1                             # rank of analysed trait in data file
WEIGHT(S)
                              # no weights
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT NESTED]
3 8 cross      # HYS, cross classified fixed effect, 3rd variable in data file, 8 levels
4 1 cov        # regression for age at calving, 4th variable in data file, 1 level
2 80 cross     # animal genetic effect, cross classified random effect, 2nd in data file, 80 levels
RANDOM_RESIDUAL VALUES
0.7                           # residual variance
RANDOM_GROUP
3                             # 3rd effect from list of effects above, with relationship
RANDOM_TYPE
add_animal                    # type of pedigree: "animal, sire, dam", missing parent = 0
FILE
rod                           # name of file with pedigree
(CO)VARIANCES
0.3                           # variance of random genetic effect animal
OPTION conv_crit 1e-17        # stopping convergence criterion
OPTION maxrounds 2000         # maxim rounds of iterations
```

Example 13. Multi-trait for variance components

Estimation of variance components by REML. Data like (Ex.7), but use MT model for 2 dependent variables "age at fist calving" and "days open". (MT calculation with much more traits was applied by Veselá et al. 2005.) Two columns are in parameter file specifying model for two traits. Both traits have the same model equation with 2 effects, fixed "HYS" and random "animal". Files are stored in ./blupf90/multi/.

Batch file remlf90.bat :

```
echo off
echo type name of parameter file
remlf90.exe > comment
echo finish of calculation
```

Parameter file paramMT:

```
# param MT
# two traits BLUP animal model
# age do = HYS + animal + e
DATAFILE
uzit                        # name of input data file
NUMBER_OF_TRAITS
2
NUMBER_OF_EFFECTS
2
OBSERVATION(S)
4 5                         # columns of analysed 2 traits (age, do) in data file
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT
NESTED]
3 3 8 cross # HYS, cross classified fixed effect, 3rd in data file, the same for 2 variables
2 2 80 cross    # animal genetic effect, 2nd in data file, for 2 variables, 80 levels
RANDOM_RESIDUAL VALUES  # (2 x 2) residual covariance matrix of expected priors
3.0 -1.1
-1.1 4.2
RANDOM_GROUP
2                           # 2nd effect from list of effects above (animal)
RANDOM_TYPE
add_animal                  # animal, parent1, parent 2
FILE                        # name of input pedigree file
rod
(CO)VARIANCES           # (2 x 2) genetic covariance matrix of expected priors
0.8 0.2
0.2 0.9
OPTION conv_crit 1e-17
OPTION maxrounds 10000
```

Example 14. GEBV with ssGBLUP

Prediction of GEBV by ssGBLUP. Data like (Ex.11), Files are stored in ./blupf90/ssgblu/. Genomic information could by store in SNP file, genomic relationship **G,** combined relationship **H** or their inversions. Here we use input of **G**. Pedigree file must consist in this case from 10 columns. The appropriate structure of columns is described together with making process in manual blupf90_all.pdf. Programs "*renumf90*" and "*pregsf90*" can be used for this case. Generally, first four columns in pedigree file are the same as in others BLUPF90 examples. Last column is original identification of animals. Columns 5th to 9th are year of birth, number of known parent, number of records for animal, number of progenies as parent 1 and number of progenies as parent 2. These

values are used for checking the consistency of data. In our example we are not using unknown parent groups neither checking of data, therefore for simplicity columns 5th to 9th are zeros.

For calculation are used 5 input files. Production records (*uzit*); pedigree (*rod2*); triangle of genomic relationship matrix (*gemat2*) with ascending renumbering of animals from 1; SNP file (*genot2*) with dense format of loci values (in our example has this file only dummy value); list of genotyped animals (*genot2_XrefID*) with original and new animal identification, name of this file corresponds with name of file with SNPs. The new options are added at the end of the parameter file (*parssg*). For genomic relationship we use weight 80% and for pedigree relationship 20% in combination into **H**. The preparation of data was performed in The SAS by "*rodssg.sas*".

Batch file blupf90.bat:

```
echo off
echo type name of parameter file
blupf90.exe > comment
echo finish of calculation
```

/*name of output commentary file from an execution of blupf90 programme

Parameters file parssg:

```
#          parssg
# single step single trait ssgBLUP animal model
# milk = HYS + age + animal + e
DATAFILE
uzit            # name of input data file
NUMBER_OF_TRAITS
1
NUMBER_OF_EFFECTS
3
OBSERVATION(S)
1               # rank of analysed trait in data file
WEIGHT(S)
                # no weights
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT NESTED]
3 8 cross       # HYS, cross classified fixed effect, 3rd variable in data file, 8 levels
4 1 cov         # regression for age at calving, 4th variable in data file, 1 level
2 80 cross # animal genetic effect, cross classified random effect, 2nd in data file, 80 levels
RANDOM_RESIDUAL VALUES
0.7             # residual variance
RANDOM_GROUP
3               # 3rd effect from list of effects above, with relationship
RANDOM_TYPE
add_animal   # type of pedigree: "animal, sire, dam", missing parent = 0
FILE
rod2            # name of file with pedigree
(CO)VARIANCES
0.3                             # variance of random genetic effect animal
OPTION SNP_file genot2          # genot2 = name of file with SNPs
OPTION saveAscii
OPTION tunedG 0
OPTION AlphaBeta 0.8 0.2    # 0.8, 0.2 = weight of genomic and pedigree relationship
OPTION readG gemat2             # gemat2 = name of file with G relationship
```

```
OPTION conv_crit 1e-17      # stopping criterion
OPTION maxrounds 2000       # maximal number of iterations
```

Example 15. RR-TDM for milk

Random regression test day model for milk (examples also in Zavadilová et al. 2005a,b and Bauer et al 2012). For each animal are with BLUP - animal model predicted "EBVs for random regression coefficients". These coefficients are subsequently used for creating EBVs of evaluated trait (milk production). Files are stored in ./blupf90/rrtd/. Legendre Polynomials (LP) with 4 terms are used for modelling of lactation curve

$$f = \mathbf{p}`\mathbf{b} \quad ,$$

where:

b = vector of regression coefficients

p = vector of parameters of the function constructed according days in milk (DIM)

Three polynomial lactation curves are included in evaluation - fixed average lactation curves for classes of effect herd, random polynomial for permanent environmental effect of each cow with production records (not correlated levels), and random polynomial for genetic effect of each animal included in pedigree file (correlated levels - relationship). 4 x 4 covariance matrices of regression coefficients within polynomial are inserted into parameter file for random effects "cow" and "animal". All polynomials use the same parameters of polynomial function. Evaluation is according to the animal model:

$$y_{ij} = HTD_i + f_{fg} + f_{pe} + f_{an} + e_{ij} \quad , \qquad (27)$$

where y_{ij} = test-day record of milk yield of cow *j* in HTD *i*;

HTD_i = herd-test-day contemporary group *i* within a herd (fixed effect);

f_{fg} = average LP of lactation curve according to fixed groups of cows within management classes of systematic environment;

f_{pe} = permanent environmental LP of lactation curve of cows, random effect with covariance matrix covering random regression coefficients;

f_{an} = genetic within lactation LP of lactation curve of animal with relationship, random effect with covariance matrix covering random regression coefficients;

e_{ij} = random residual of test day record, reflecting changes of variance along the course of lactation. Residual variance is used for creating a weight for weighted analysis.

Data for test days are artificially extended from lactations in (Ex 7.) using programmes "*tvor.sas*". Input raw data are in file *prodrec.prn* and have following structure:

Herd	Cow	Date test day	Date calving	Milk/day
1	11	28 5 2010	12 5 2010	19
		28 6 2010		29
		28 7 2010		37
		28 8 2010		40
		28 9 2010		37
		28 10 2010		32
		28 11 2010		31
		28 12 2010		28
1	12	28 9 2010	8 9 2010	18
		28 10 2010		30

Unknown sires are located in one “unknown parent group” and unknown mothers in two “unknown parent groups”, pedigree file has 4 columns (animal, parent1, parent2, coefficient). The file with effects for BLUP evaluation is created in The SAS by the “*prepar.sas*”.

```
/*.....................................................prepar.sas.....................................................*/
/*    preparing production file for RR TD model of milk   */
filename   prod    "c:\LinMod\blupf90/rrtd/prodrec.prn";  /* raw input file */
filename   record  "c:\LinMod\blupf90/rrtd/record";              /* file for calculation */
data raw ;
      infile prod ;
      input herd anim dayr monthr yearr dayc monthc yearc milk ;
      drec = dayr + (monthr-1)*30 + (yearr-1)*365 ;         /*day of recording */
      dbic = dayc + (monthc-1)*30 + (yearc-1)*365 ;         /*day of calving */
      dim = drec - dbic ;                                   /* days in milk */
      htd = compress(herd|| dayr||monthr||yearr);           /* herd-test-day*/
proc means ;
            /*.................................................recoding ..................................................*/
proc sort ; by htd;                          /* recoding HTDs from 1 */
data a ;
      set raw ; by htd; if first.htd ;
      keep htd ;
data b ;                                      /* new code list of HTD */
      set a ;
      nh = _n_ ;
proc print ;
data raw2 ;
      merge raw b ; by htd ;
      keep herd nh anim dim milk ;
proc means ;
proc sort ; by anim;                         /* recoding cows from 1 */
data a ;
      set raw2 ; by anim; if first.anim ;
      keep anim ;
data b ;                                     /* new code list of cows */
      set a ;
      cow = _n_ ;
proc print ;
data raw3 ;
      merge raw2 b ; by anim ;
      keep herd nh cow anim dim milk ;
proc means ;
proc freq; tables nh ;
          /* .....................................parameters for LP regressions ..................... */
data regrcov ;
      set raw3 ;
      sv = 2*((dim-1)/305)-1;
      p1 = sv*sqrt(3);                    p1 = round(p1, .00001);
      p2 = 0.5*(3*sv*sv-1)*sqrt(5);   p2 = round(p2, .00001);
      p3 = 0.5*(5*sv**3-3*sv)*sqrt(7); p3 = round(p3, .00001);
      if p1 = 0 then p1= 0.00001 ;
      if p2 = 0 then p2= 0.00001 ;
```

```
      if p3 = 0 then p3= 0.00001 ;
/*... residual variances according the parts of lactation Zavadilova et al., 2005).....*/
      v1=8.1205614; v2=4.9632274; v3=3.9800503; v4=4.1415464;
      vr = (45*v1+70*v2+150*v3+40*v4)/305; /*average residual variance*/
            /*.............................weights according parts of lactation.....................*/
      if dim <  46 then weight = vr/v1 ;
      else if dim < 116 then weight = vr/v2 ;
      else if dim < 266 then weight = vr/v3 ;
      else weight = vr/v4 ;
      weight = round(weight, .00001);
      file record ;
      put herd nh cow anim milk p1 p2 p3 dim weight ;
proc means;
proc print ;
run;/*.............................finish.................*/
```

Batch file bluplf90.bat :

```
echo off
echo type name of parameter file
blupf90.exe > comment
echo finish of calculation
```

Parameter file paramRR:

```
#           param RR
# random regression TD model
# milk = HTD + fix reg (within herd)
#          + random reg PE (within cow)
#          + random genetic reg (within animal) + e
DATAFILE
record
NUMBER_OF_TRAITS
1
NUMBER_OF_EFFECTS
13
OBSERVATION(S)
5                    # column of analysed trait in data file
WEIGHT(S)
10                   # column with weight in data file
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT
NESTED]
2 24 cross     # HTD, cross classified fixed effect, 2nd in data file, 24 levels
1 2 cross      # herd, 1st in data file, fixed effect, 2 levels
6 2 cov 1      # 1st parameter for regression of lactation curve, fixed effect nested within 1
7 2 cov 1      # 2nd parameter for regression of lactation curve, fixed effect nested within 1
8 2 cov 1      # 3rd parameter for regression of lactation curve, fixed effect nested within 1
3 59 cross     # cow permanent environment (PE), random effect, 59 levels
6 59 cov 3     # 1st parameter for regression of lactation curve, random effect nested within 3
7 59 cov 3     # 2nd parameter for regression of lactation curve, random effect nested within 3
```

```
8 59 cov 3    # 3rd parameter for regression of lactation curve, random effect nested within 3
4 83 cross    # animal genetic, random effect, 83 levels
6 83 cov 4    # 1st parameter for regression of lactation curve, random effect nested within 4
7 83 cov 4    # 2nd parameter for regression of lactation curve, random effect nested within 4
8 83 cov 4    # 3rd parameter for regression of lactation curve, random effect nested within 4
RANDOM_RESIDUAL VALUES
4.58820       # average residual variance
RANDOM_GROUP      # 1st random group
6 7 8 9       # rank of correlated effects for cow PE with conjoint covariance matrix
RANDOM_TYPE       # no relationship
diagonal
FILE

(CO)VARIANCES     # (4 x 4) PE covariance matrix of regression coefficients
    6.8489355   0.3630769   -0.075673   -0.061666
    0.3630769   1.5650312   0.1232025   -0.071714
    -0.075673    0.1232025   0.5213394   0.029643
    -0.061666   -0.071714    0.029643    0.2435163
RANDOM_GROUP  # 2nd random group
10 11 12 13           # rank of correlated effects for animal with conjoint covariance matrix
RANDOM_TYPE       # type of relationship with phantom parents group
add_an_upg
FILE                  # name of pedigree file
rod3
(CO)VARIANCES     # (4 x 4) genetic covariance matrix of regression coefficients
    3.3896411  0.3046061  -0.479661    0.176901
    0.3046061  0.4755081   0.0248085   0.0016134
   -0.479661    0.0248085   0.3157532  -0.104984
    0.176901    0.0016134  -0.104984    0.0739097
OPTION conv_crit 1e-17
OPTION maxrounds 10000
```

Example 16. EBV for direct and maternal genetic effects

Prediction of EBVs for genetically correlated direct and maternal effects (examples also in Přibyl et al. 2003 and Vostrý et al 2012). Files are stored in ./blupf90/matblu/. Analysed trait is yearlings live weight in a suckle calf system.

Data are related:

Sire (1) is sire only of mothers (4, 5, 6, 7).
Sire (2) is sire of mothers (8, 9, 10) and calves with performance record (11, 12, 13, 14).
Sire (3) is sire only of calves with performance record (15, 16, 17, 18, 19, 20, 21).
Cows (8, 9) have also performance record as calves.
Cows (9, 10) have each only one progeny with performance record.
Cows (4, 6, 7, 8) have each two progenies with performance record.
Cow (5) has three progenies with performance record.

Two correlated genetic effects covered by conjunct covariance matrix influence result - growth ability of calf and maternal ability of mother. Together with maternal permanent environment are in evaluation three random effects. Nature of the system is possible to describe by model equation:

$$y_{ijklmn} = HYS_i + sex_j + anim_k + gmat_l + emat_m + e_{ijklmn} \quad , \qquad (28)$$

where: y_{ijklmn} = yearlings live weight;
HYS_i = herd-year-season classes of contemporary groups (fixed effect);

sex_j = sex of calf (fixed effect);
$anim_k$ = animal direct genetic (random effect);
$gmat_l$ = maternal genetic (random effect);
$emat_m$ = maternal permanent environment (random effect);
e_{ijklmn} = random residual.

Batch file bluplf90.bat :

```
echo off
echo type name of parameter file
blupf90.exe > comment
echo finish of calculation
```

Parameter file parmat:

```
#          parmat
# BLUP- maternal animal model
# y_ijklmn = HYS_i + sex_j + anim_k + gmat_l + emat_m + e_ijklmn
DATAFILE
gro
NUMBER_OF_TRAITS
1
NUMBER_OF_EFFECTS
5
OBSERVATION(S)
6              # column of analysed trait in data file
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT
NESTED]
4 2 cross      # HYS, cross classified fixed effect, 4th in data file, 2 levels
5 2 cross      # sex of calf, 5th in data file, fixed effect, 2 levels
1 21 cross     # animal direct genetic, 1st in data file, random with relationship
2 21 cross     # maternal genetic, 2nd in data file, random with relationship
3 8 cross      # maternal permanent environment, 3rd in data file, random diagonal
RANDOM_RESIDUAL VALUES
1154                    # residual variance
RANDOM_GROUP  # 1st random group
5                       # rank of effect for cow PE with variance
RANDOM_TYPE     # no relationship for PE
diagonal
FILE
(CO)VARIANCES    # PE variance
    86
RANDOM_GROUP   # 2nd random group
3 4         # rank of correlated effects for animal with conjoint genetic covariance matrix
RANDOM_TYPE      # type of relationship
add_animal
FILE                    # name of pedigree file
matped
(CO)VARIANCES    # (2 x 2) genetic covariance matrix of direct and maternal effect
```

```
    692  -49
    -49  107
OPTION conv_crit 1e-17
OPTION maxrounds 10000
```

Example 17. ssGBLUP for RR-TDM with three lactations

Prediction of GEBVs by ssGBLUP for test-days in three lactations. Data from Ex.11 and 14 are extended to test-day records and three lactations (was done together with Ex. 15). Number of observation decreases with the age of cows. Regression coefficients for random polynomials are dependent. PE and genetic covariance matrices are (12 x 12) and cover all regressions in three lactations. Blocks of elements in covariance matrices are ordered according effects; within block are all three traits. Files are stored in ./blupf90/ssg3lb/.

Evaluation is according to the three-lactation test day animal model with 4-parameter Legendre Polynomials (LP). Model equation for the 1st lactation is different form lactation 2nd and 3rd:

$$y_{ijn} = HTD_{in} + \beta_1 \cdot ca_j + \beta_2 \cdot ca_j^2 + \beta_3 \cdot do_{jn} + \beta_4 \cdot do_{jn}^2 + \beta_5 \cdot ci_{jn} + \beta_6 \cdot ci_{jn}^2 + f_{fg,n} + f_{pe,n} + f_{an,n} + e_{ijn} \quad , \qquad (29)$$

where y_{ijn} = test-day record of milk yield of cow in lactation *n* <1,2,3>;
HTD_{in} = herd-test-day-parity contemporary group *i* within a herd in lactation *n*, fixed effect;
β_1, β_2, β_3, β_4 β_5 and β_6= fixed regression coefficients;
ca_j and ca_j^2 = parameters for curvilinear regressions on calving age for 1st lactation, fixed effect;
do_{jn} and do_{jn}^2 = parameters for curvilinear regressions on days open within current lactation, fixed effect;
ci_{jn} and ci_{jn}^2 = parameters for curvilinear regressions on previous calving interval for 2nd and 3rd lactations, fixed effect;
$f_{fg,n}$ = average LP of lactation curve according to groups of cows within management classes of systematic environment (herd x parity), fixed effect;
$f_{pe,n}$ = permanent environmental within lactation LP of lactation curve of cows, random effect with covariance matrix (Zavadilová et al., 2005a;b);
$f_{an,n}$ = genetic within lactation LP of lactation curve of animal, random effect with covariance matrix and relationship;
e_{ijn} = random residual of test day records within lactation *n*, reflecting changes of variability along the course of lactation (modelled by weighed analysis).

Pedigree file *rod2* (without phantom parent groups) is the same like in Ex. 14 with 10 columns. File of production records *uzit3ss* contains 18 columns: herd-test-day classes, herd, cow, animal, test-day milk1, milk2, milk3, parameters for ca, ca^2, do, do^2, ci, ci^2, parameters for lp1, lp2, lp3, DIM, weights. Missing values = 0. Input of genomic information is through **G** matrix. For calculation are used 5 input files. Production records (*uzit3ss*); pedigree (*rod2*); triangle of genomic relationship matrix (*gemat2*) with ascending renumbering of animals from 1; SNP file (*genot2*) with dense format of loci values; list of genotyped animals (*genot2_XrefID*) with original and new animal identification, name of this file corresponds with name of file with SNPs. To the parameter file (*parRR3lg*) are added on the end options. Fore genomic relationship we use weight 80% and for pedigree relationship 20% in combination into **H**.

With parameter file *parRR3lg* is running calculation of GEBV and with parameter file *parRR3l* (in directory) model for usual EBV without genomic. Three columns are in parameter file specifying different model equation for three traits (0= missing effect).

Batch file bluplf90.bat :

```
echo off
echo type name of parameter file
blupf90.exe > comment
echo finish of calculation
```

Parameter file parRR3lg:

```
#          paraRR3lg
# random regression TD model for 3 lactations genomic
# milk = HTD + fixed effects + fix reg (within herd x parity)
#          + random reg PE (within cow)
#          + random genetic reg (within animal) + e
DATAFILE
uzit3ss
NUMBER_OF_TRAITS
3
NUMBER_OF_EFFECTS
19
OBSERVATION(S)
5 6 7                    # column of three analysed trait in data file
WEIGHT(S)
18                       # column with weight in data file
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT [EFFECT
NESTED]
 1   1   1 68 cross      # HTD, cross classified fixed effect, 1st in data file, 68 levels
 2   2   2   2 cross      # herd-parity, cross classified fixed effect, 2nd in data file, 2 levels
 8   0   0   1 cov        # linear regres of age at calving in first lactation, fixed effect, 1 level
 9   0   0   1 cov        # quadratic regres of age at calving in first lactation, fixed effect, 1 level
10 10 10   1 cov        # linear regres of days open in all three lactations, fixed effect, 1 level
11 11 11   1 cov        # quadratic regres of days open in three lactations, fixed effect, 1 level
 0 12 12   1 cov        # linear regres of calving interval in 2nd and 3rd lactation, fixed effect
 0 13 13   1 cov        # quadrat regres of calving interval in 2nd and 3rd lactation, fixed effect
14 14 14   2 cov 2 2 2 # 1st parameter for regres of lact. curve, fixed effect nested within 2
15 15 15   2 cov 2 2 2 # 2nd parameter for regres of lact. curve, fixed effect nested within 2
16 16 16   2 cov 2 2 2 # 3rd parameter for regres of lact. curve, fixed effect nested within 2
 3   3   3 59 cross       # cow permanent environment (PE), random effect, 59 levels
14 14 14 59 cov 3 3 3 # 1st parameter for random regres of lactation curve, nested within 3
15 15 15 59 cov 3 3 3 # 2nd parameter for random regres of lactation curve, nested within 3
16 16 16 59 cov 3 3 3 # 3rd parameter for random regres of lactation curve, nested within 3
 4   4   4 80 cross       # animal genetic, random effect, 80 levels
14 14 14 80 cov 4 4 4 # 1st parameter for random regres of lactation curve, nested within 4
15 15 15 80 cov 4 4 4 # 2nd parameter for random regres of lactation curve, nested within 4
16 16 16 80 cov 4 4 4 # 3rd parameter for random regres of lactation curve, nested within 4
RANDOM_RESIDUAL VALUES
4.84  0    0                # (3 x 3) residual covariance matrix
0   7.36   0
0     0  8.57
RANDOM_GROUP    # 1st random group
12 13 14 15              # rank of correlated effects for cow PE with conjoint covariance matrix
```

```
RANDOM_TYPE      # no relationship
diagonal
FILE

(CO)VARIANCES     # (12 x 12) PE covariance matrix of regression coefficients
6.8489355 3.7511534 3.2634698 0.3630769 0.1681511 0.2150463 -0.075673 -0.101992 0.0275057
-0.061666 -0.05431 -0.081858
3.7511534 11.17252 6.1256801 0.7428758 0.6147304 0.1941889 0.0136907 -0.457097
-0.211333 -0.002865 -0.101704 -0.067442
3.2634698 6.1256801 12.923124 0.4441378 1.0718576 0.4005584 0.0086804 -0.183431
-0.489477 -0.094197 -0.146208 -0.059229
0.3630769 0.7428758 0.4441378 1.5650312 0.2836088 0.177847 0.1232025 -0.092613
-0.01371 -0.071714 -0.002055 -0.062981
0.1681511 0.6147304 1.0718576 0.2836088 2.7287618 0.7177153 -0.048012 0.0952177
-0.028853 0.0005802 -0.191659 -0.181254
0.2150463 0.1941889 0.4005584 0.177847 0.7177153 3.0229527 -0.053467 0.0662482 0.1495489
0.0074918 -0.141721 -0.214024
-0.075673 0.0136907 0.0086804 0.1232025 -0.048012 -0.053467 0.5213394 0.0438634 0.0151852
0.029643 0.0091667 0.0019495
-0.101992 -0.457097 -0.183431 -0.092613 0.0952177 0.0662482 0.0438634 0.9689733 0.2556847
0.0110097 -0.026758 -0.022478
0.0275057 -0.211333 -0.489477 -0.01371 -0.028853 0.1495489 0.0151852 0.2556847 1.1552266
0.0096294 -0.010958 -0.076793
-0.061666 -0.002865 -0.094197 -0.071714 0.0005802 0.0074918 0.029643 0.0110097 0.0096294
0.2435163 0.0179287 0.0083403
-0.05431 -0.101704 -0.146208 -0.002055 -0.191659 -0.141721 0.0091667 -0.026758
-0.010958 0.0179287 0.3700977 0.0618851
-0.081858 -0.067442 -0.059229 -0.062981 -0.181254 -0.214024 0.0019495 -0.022478
-0.076793 0.0083403 0.0618851 0.399678
RANDOM_GROUP      # 2nd random group
16 17 18 19            # rank of correlated effects for animal with conjoint covariance matrix
RANDOM_TYPE   # type of relationship
add_animal
FILE                  # name of pedigree file
rod2
(CO)VARIANCES  # (12 x 12) genetic covariance matrix of regression coefficients
3.3896411  3.60412  3.4580753  0.3046061  -0.093063  0.1170898  -0.479661  -0.320409
 -0.486612 0.176901 0.1183183 0.1616494
3.60412 4.8106244 4.5717311 0.4816699 0.199605 0.4481552 -0.493112 -0.340016
-0.482291 0.1853526 0.112859 0.1605075
3.4580753 4.5717311 5.3210489 0.386692 0.201976 0.3716594 -0.515037 -0.404006
-0.504233 0.1885794 0.1031577 0.1570471
0.3046061 0.4816699 0.386692 0.4755081 0.6455554 0.7235655 0.0248085 0.0783388 0.0386035
0.0016134 0.009996 -0.016537
-0.093063 0.199605 0.201976 0.6455554 1.7407905 1.7462697 0.3191798 0.3766198 0.409802 -
0.11972 -0.112999 -0.150833
0.1170898 0.4481552 0.3716594 0.7235655 1.7462697 2.1440625 0.3137405 0.415393 0.4336006
-0.094165 -0.096003 -0.126288
-0.479661 -0.493112 -0.515037 0.0248085 0.3191798 0.3137405 0.3157532 0.2878517 0.2798391
-0.104984 -0.092611 -0.110833
-0.320409 -0.340016 -0.404006 0.0783388 0.3766198 0.415393 0.2878517 0.3786789 0.3209385 -
0.080907 -0.07372 -0.129177
```

-0.486612 -0.482291 -0.504233 0.0386035 0.409802 0.4336006 0.2798391 0.3209385 0.5252568 -0.086951 -0.089642 -0.152298
0.176901 0.1853526 0.1885794 0.0016134 -0.11972 -0.094165 -0.104984 -0.080907 -0.086951 0.0739097 0.0574635 0.0504514
0.1183183 0.112859 0.1031577 0.009996 -0.112999 -0.096003 -0.092611 -0.07372 -0.089642 0.0574635 0.0959358 0.0713992
0.1616494 0.1605075 0.1570471 -0.016537 -0.150833 -0.126288 -0.110833 -0.129177 -0.152298 0.0504514 0.0713992 0.1654504
OPTION SNP_file genot2 # genot2 = name of file with SNPs
OPTION saveAscii
OPTION tunedG 0
OPTION AlphaBeta 0.8 0.2 # 0.8, 0.2 = weight of genomic and pedigree relationhip
OPTION readG gemat2 # gemat2 = name of file with G relationship
OPTION conv_crit 1e-17 # stopping criterion
OPTION maxrounds 20000 # maximal number of iterations

6. DMU programs

The DMU package (Madsen et al., 2010) is used to estimation of variance components and solving of mixed models by several methodologies. There are a several modules. The module *dmu1* is executed automatically as initial step with all calculations. Package was developed for Linux and adapted for others operation systems. Distributions and documentation are on http://dmu.agrsci.dk . Basic manual is dmuv6_guide.5.5.pdf.

Mandatory files used for calculations are - *data*, *pedigree*, *run_dmu* script and *parameters* (directive file) with extension *.DIR*. Data could be in ascii or binary form. Columns in a data file must be arranged as follows: first had to be columns intended for variables in the integer format (e.g. identifications of herds, animals), followed by columns with variables in the real format (e.g. dependent variables and covariables - regressions). There are four columns – animal, sire, dam, and birth (ascendants) sequence in the pedigree file. Phantom parents groups are coded with negative values, if they occurred. Parameters could be located in parameter file or read from several external files. Results are located in file with extension *.lst* and *.SOL*. There are eight basic columns (type of effect, trait, random effect number, effect within submodel, level, number of observation in class, consecutive class number and solution value) in *.SOL* file.

Additional attached programs, for example *DmuTrace* and *G-matrix* (Su and Madsen 2011), could be used for preparing and checking consistency of data files. Useful is the interface with free R-project software, with which could work interactively.

Running under Linux

Program is located in the directory */user/local/bin*. Into your directory with data-files locate the start-up file according module with which you will do the calculation (*r_dmuxx* script (for example *r_dmu5*)) and parameter file with extension .DIR (for example *sindmu5.DIR*). Calculation is executed by submitting from the command line the command: *nohup ./r_dmu5 sindmu5 &* . Results are stored (according specification) in *run_dmuxx* located in files *sindmu5.lst* and *sindmu5.SOL*.

Example of file r_dmu5

```
#!/bin/bash
if [  $# -eq 0 ]
  then
    name=test5
  else
```

```
   name=$1
fi

export name
time dmu1 < $name.DIR > $name.lst
if [ -s MODINF ]
 then                                          # specification of module for BLUP
   echo '1' >> $name.lst
 time dmu5 >> $name.lst
fi
rm -f CODE_TABLE DMU_LOG fort.71 fort.70
rm -f DUMMY MODINF DMU1.dir DMU5.dir PARIN
rm -f RCDATA_I RCDATA_R
rm -f PEDFILE* AINV* fort.* ]

if [ -f INBREED   then
   if [ -s INBREED ]
     then
       mv INBREED $name.INBREED
    else
      rm INBREED
   fi
fi
if [ -s SOL ]
 then
 mv SOL $name.SOL
  cmp -s  $name.SOL $name.SOL.org
  if [ $? -eq 0 ]
    then
      echo "Example $name in $PWD OK" >> ../run_ex.log
    else
      echo "Example $name in $PWD failed - Check output files" >> ../run_ex.log
    fi
fi
```

Running under Windows

Windows version is usually installed into C:\Program Files\QGG-AU\DMUv6\, where also examples can be found. Within, in a subdirectory “bin” is located “DMU.bat” file. In our case has form:

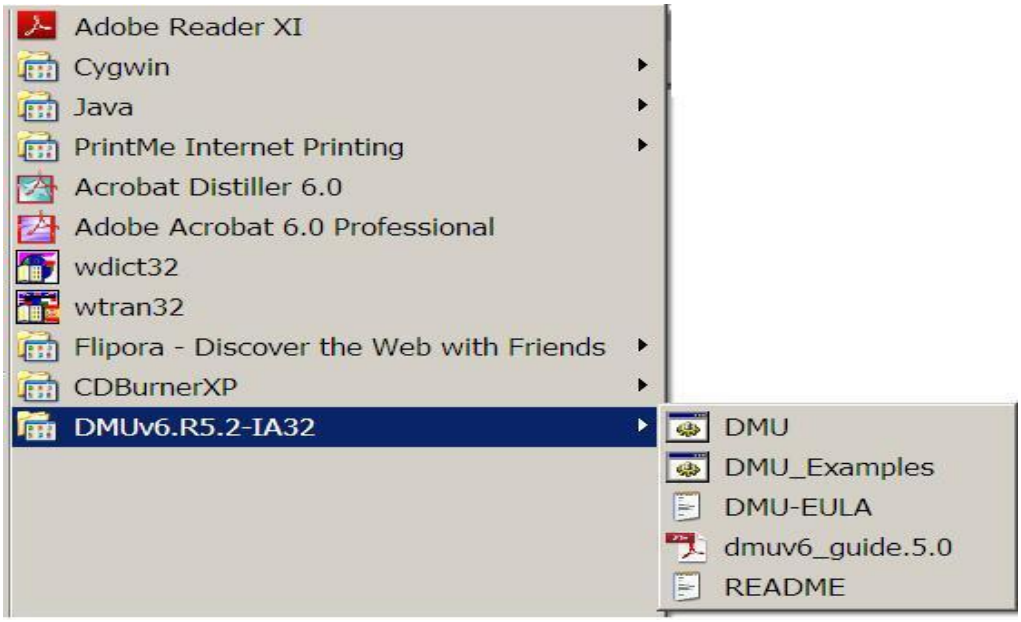
```
cmd.exe /T:70 /D /Q /K "cd C:\ && set PATH=C:\Program Files\QGG-AU\DMUv6\R5.2\bin;%PATH% &&
TITLE DMU && mode con lines=65 cols=125 && echo. && echo   You can now change to the directory
where you want to run DMU && echo. "
```

The DMU can be accessed through Start -> All programs (Programs) -> DMU, or by right mouse button menu.

The DMU entry opens a consol window for running the *run_dmuxx* scripts for your analysis. Basic DOS command, like cd , cd .. , copy, del, dir, edit, exit, md, print, set, ... are useful for working within a window. It is possible to go into your directory with data and submit calculation. The syntax for the *run_dmuxx* script is:

run_xxxx filename

where: *xxxx* is *dmu4*, *dmu5*, *dmuai* or *rjmc*,
filename is name of directing parameter file, located in your current directory with the extension *.DIR*.



Example 18. ST animal model

Like (Ex.7, 12), BLUP prediction of EBV with module DMU4 or DMU5. Files are stored in ./LinMod/dmu/sin/. Prior known heritability of analysed trait is 0.30. To run the program, write the: *run_dmu4 sindmu5*.
Production file is rearranged by SAS programme:

convprod.sas

```
filename star "c:/LinMod/dmu/sin/uzit";
filename nov "c:/LinMod/dmu/sin/uzit2";
data a;
     infile star;
     input milk anim herd age dopen ;
     age = age - 27 ; /* standardization of age */
     file nov ;
     put herd anim milk age dopen ;
run;
```

Directing parameter file sindmu5.DIR (when copying into calculation eliminate remarks)

```
$COMMENT
prediction EBV with DMU5 (put 12) or for DMU4 (put 11)
$ANALYSE 11 2 0 0        # 12(11)= BLUP with DMU5(DMU4), 2= method PCG (JSI),
                         # 0= no scaling, 0= minim output
$DATA  ASCII (2, 3, -999) uzit2   # 2 integer, 3 real variables, value missing = -999,
                                  # name of input production file "uzit2"
$VARIABLE
herd anim milk age dopen          # sequence of 5 variables in input file

$MODEL
1                                 # 1 analysed trait "milk"
0                                 # no absorption
1 0 2 1 2                         #1=first analysed trait, 0=no weight, 2= two effects in classes,
                        # 1= position of fixed effect herd, 2= position of random effect anim
```

```
1 1                  #  1= one random effect, 1= first random effect anim
1 2                  #   = one regression, 2 = regression is second real variable
0
$VAR_STR 1 PED 1 ASCII rod4   # 1 = first random effect, PED = type of relationship,
                     # 1 = sire+dam+inbreeding, rod4= name of pedigree file
$PRIOR
1  1  1  0.30        # 1 = covariance matrix (1x1) for first random effect animal
2  1  1  0.70        # 2 = covariance matrix (1x1) residual (last random effect)
$DMU5                # options for DMU5
30000 0.1E-11        # maximum no. of iterations, finishing convergence
```

Example 19. Variance components for MT

Like (Ex.13). Estimation of variance components by REML with module DMUAI. Files are stored in ./LinMod/dmu/mult/. To run the program, write the: *run_dmuai multai*.

Directing parameter file multai.DIR (when copying into calculation eliminate remarks)

```
$COMMENT
variance components with DMUAI
$ANALYSE 1 2 0 0          # 1= REML with DMUai, 2= method EM, 0= no scaling,
                          #   0= minim output
$DATA  ASCII (2,3,-999) uzit2     #2 integer, 3 real variables, value missing = -999,
                                  # uzit2 = name of production file
$VARIABLE
herd anim milk age dopen          # sequence of 5 variables in input file

$MODEL
2                                 # 2 analysed traits
0                                 # no absorption for 1st trait
0                                 # no absorption for 2nd trait
2 0 2 1 2                   #2= 1st analysed trait, 0=no weight, 2= two effects in classes,
3 0 2 1 2                   #3= 2nd analysed trait, 0=no weight, 2= two effects in classes,
1 1                         # for 1st trait, 1= one random effect, 1= first random effect anim
1 1                         # for 2nd trait, 1= one random effect, 1= first random effect anim
0                           # no regression for 1st trait
0                           # no regression for 2nd trait
0
$VAR_STR 1 PED 1 ASCII rod4   #1 = first random effect, PED = type of relationship,
                     #1 = sire+dam+inbreeding, rod4= name of pedigree file
$PRIOR
1  1  1  0.80        # 1 = triangle of prior cov. matrix (2x2) for first random effect animal
1  2  1  0.20
1  2  2  0.90
2  1  1  3.00        # 2 =triangle of prior covariance matrix (2x2) residual
2  2  1  1.10
2  2  2  4.20

$SOLUTION            # 0= time optimized of FSPAK
```

Example 20. GEBV with ssGBLUP method

Like (Ex.11, 14), prediction of GEBV by ssGBLUP. Mayor part of parameters the same like in (Ex.16). Inputs into calculation are 4 files - production *records*, *pedigree*, *G-matrix*, and *list of genotyped* animal. Files are stored in ./LinMod/dmu/ssg/. Calculation submitted by writing: *run_dmu4 ssgdmu5*.

Directing parameter file ssgdmu5.DIR (when copying into calculation eliminate remarks)

```
$COMMENT
ssGEBV prediction of GEBV with DMU5 or DMU4
$ANALYSE 11 2 0 0          # 12(11)= BLUP with DMU5(DMU4), 2= method PCG (JSI),
                           # 0= no scaling, 0= minim output
$DATA  ASCII (2,3,-999) uzit2    # 2 integer, 3 real variables, value missing = -999,
                                 # uzit2 =  name of production file
$VARIABLE
herd anim milk age dopen         # sequence of 5 variables in input file

$MODEL
1                                # 1 analysed trait "milk"
0
1 0 2 1 2              #1=first analysed trait "milk", 0=no weight, 2= two effects in classes,
                       #1= position of fixed effect herd, 2= position of random effect anim
1 1                    # 1= 1 random effect, 1= first random effect anim
1 2                    # 1 = one regression, 2 = regression is second real variable "age"
0
$VAR_STR 1 PGMIX 2 ASCII rod4 sezgenot gemat 0.20 #1 = first random effect, PGMIX
        # = type of combined relationship, 2 = sire+dam, rod4= name of pedigree file,
        # sezgenot= name of file of genotyped animals, gemat= file with genomic
        # relationship, 0.20= weight of pedigree relationship
$PRIOR
1 1  1  0.30           # 1 = covariance matrix (1x1) for first random effect
2  1  1  0.70          # 2 = covariance matrix (1x1) residual

$DMU5                  # options for DMU5
30000 0.1E-11          # maximum no. of iterations, finishing convergence
```

Example 21. RR-TDM for milk

Like (Ex.15), RR-TDM with DMU5(DMU4). Phantom parents groups are marked in pedigree file with negative value. Files are stored in ./LinMod/dmu/rrdmu/. Calculation submitted by writing: *run_dmu4 rrdmu5*.

Directing parameter file rrdmu5.DIR (when copying into calculation eliminate remarks)

```
$COMMENT
RR TDM with DMU5(12) or DMU4(11)
$ANALYSE 11 2 0 0          # 12(11)= BLUP with DMU5(DMU4), 2= method PCG (JSI),
                           # 0= no scaling, 0= minim output
```

```
$DATA  ASCII (4,6,-999) record    # 4 integer, 6 real variables, value missing = -999,
                                  # record =name of production file
$VARIABLE
herd HTD cow anim milk lp1 lp2 lp3 dim weight      # sequence of 10 variables in input file

$MODEL
1                                 # 1 analysed trait
0
1 6 4 2 1 3 4                     #1=first analysed trait, 6=weight 6th real, 4= four effects in classes,
                                  # 2= position of fixed effect HTD, 1= position of fixed effect herd,
                                  # 3 position of random effect cow, 4= position of random effect animal
2  1 2                            # 2= two random effects, 1= cow, 2=animal
9 2(2 3 4) 3(2 3 4) 4(2 3 4)      # 9= nine regressions, 2,3,4 (lp1, lp2, lp3) each nested within
                                  # effects 2,3,4 (herd, cow, animal)
0
$VAR_STR 2 PED 6 ASCII rod3d   # 2=second random effect animal, PED=type of relat,
                  # 6 = +sire+dam+phantom parents groups, rod3d= name of pedigree file
$PRIOR
1 1 1  6.8489355        # 1 = triangle of permanent environment covariance matrix (4x4) for
1 2 1  0.3630769        #first random effect cow
1 2 2  1.5650312
1 3 1 -0.075673
1 3 2  0.1232025
1 3 3  0.5213394
1 4 1 -0.061666
1 4 2 -0.071714
1 4 3  0.029643
1 4 4  0.2435163
2 1 1  3.3896411        # 2 = triangle of genetic covariance matrix (4x4) for second random
2 2 1  0.3046061        #effect animal
2 2 2  0.4755081
2 3 1 -0.479661
2 3 2  0.0248085
2 3 3  0.3157532
2 4 1  0.176901
2 4 2  0.0016134
2 4 3 -0.104984
2 4 4  0.0739097
3 1 1  4.58820          # 3 = covariance matrix (1x1) for last random effect residual

$DMU5
30000 0.1E-11
```

Example 22. EBV for direct and maternal genetic effects

Like (Ex. 16). Files are stored in ./dmu/matdm/. Calculation submitted by writing: *run_dmu5 matdmu5*.

Directing parameter file matdmu5.DIR (when copying into calculation eliminate remarks)

```
$COMMENT
Maternal for growth with DMU5(12) or DMU4(11)
```

```
$ANALYSE 11 2 0 0              # 12(11)= BLUP with DMU5(DMU4), 2= method PCG (JSI),
                               # 0= no scaling, 0= minim output
$DATA  ASCII (5,1,-999) gro    # 5 integer, 1 real variables, value missing = -999,
                               # gro =name of production file
$VARIABLE
anim gmat emat HYS sex livweight # sequence of 6 variables in input file

$MODEL
1                                      # 1 analysed trait
0
1  0 5 4 5 3 1 2     #1=first analysed trait, 0= no weight, 5= five effects in classes,
                     # 4= position of fixed effect HYS, 5= position of fixed effect sex,
                     # 3 position of random effect cow PE, 1= position of direct genetic
                     # random effect, 2= position of maternal genetic random effect
3 1 2 2              # 3= three random effects, 1= 1st random PE, 2= 2nd random animal direct,
                     # 2= 3rd random genetic maternal (also animal)
0                    # no regressions
0
$VAR_STR 2 PED 2 ASCII matped  # 2=second random effect animal direct,
                               # PED=type of relationship,
                               # 2 = animal+sire+dam+no inbreeding,
                               # matped= name of pedigree file

$PRIOR
1 1 1     86          # 1 =  permanent environment covariance matrix (1x1)
2 1 1    692          # 2 = triangle of genetic direct with maternal covariance matrix (2x2)
2 2 1     -49
2 2 2    107
3 1 1  1154           # 3 = covariance matrix (1x1) for last random effect residual

$DMU5
30000 0.1E-11
```

Example 23. GEBV for RR-TDM with three lactations

Like (Ex. 17). Files are stored in ./dmu/ssg3lb/. Blocks of elements in covariance matrices are ordered according traits; within trait are covariances between regression coefficients for polynomials. Calculation is submitted by writing: *run_dmu5 rr3ldmg* for GEBV and *run_dmu5 rr3ldm* (in directory) for usual *EBV*.

Directing parameter file rr3ldmg.DIR (when copying into calculation eliminate remarks)

```
$COMMENT
RR TDM for 3 lactations with DMU5 or DMU4
herd-test-day classes, herd, cow, animal, test-day milk1, milk2, milk3,
parameters for ca, ca2, do, do2, ci, ci2, parameters for lp1, lp2, lp3, DIM, weights.
Missing values = -999 ;
$ANALYSE 11 2 0 0          # 12(11)= BLUP with DMU5(DMU4), 2= method PCG (JSI),
                           # 0= no scaling, 0= minim output
$DATA  ASCII (4,14,-999) uzit3ss # 4 integer, 14 real variables, value missing = -999,
                           # uzit3ss =name of production file
$VARIABLE                  # sequence of 18 variables in input file
```

```
HTD hl cow anim milk1 milk2 milk3 ca ca2 do do2 ci, ci2 lp1 lp2 lp3 dim weight
$MODEL
3                                   # 3 analysed traits
0
0
0
1 14 4 1 2 3 4      #1=first analysed trait, 14= position of weight, 4= four effects in classes,
                    # 1= position of fixed effect HTD, 2= position of fixed effect herd*lact,
                    # 3 position of random effect cow, 4= position of random effect animal
2 14 4 1 2 3 4      #2=second trait, 14= position of weight, 4= four effects in classes,
                    # 1= position of fixed effect HTD, 2= position of fixed effect herd*lact,
                    # 3 position of random effect cow, 4= position of random effect animal
3 14 4 1 2 3 4      #3=third trait, 14= position of weight, 4= four effects in classes,
                    # 1= position of fixed effect HTD, 2= position of fixed effect herd*lact,
                    # 3 position of random effect cow, 4= position of random effect animal
2 1 2               # for 1st trait, 2= two random effects, 1= random cow, 2= random anim
2 1 2               # for 2nd trait, 2= two random effects, 1= random cow, 2= random anim
2 1 2               # for 3rd trait, 2= two random effects, 1= random cow, 2= random anim
13 4 5 6 7 10(2 3 4) 11(2 3 4) 12(2 3 4) # 13= 13 regressions for first trait,
                                         # 4,5,6,7 regression for age and days open,
                                         # 10,11,12  Leg. Polynom within herd, cow, anim
13 6 7 8 9 10(2 3 4) 11(2 3 4) 12(2 3 4) # 13= 13 regressions for second trait,
                                         # 6,7,8,9  regression for days open, calving interval
                                         # 10,11,12  Leg. Polynom within herd, cow, anim
13 6 7 8 9 10(2 3 4) 11(2 3 4) 12(2 3 4) # 13= 13 regressions for third trait,
                                         # 6,7,8,9  regression for days open, calving interval
                                         # 10,11,12  Leg. Polynom within herd, cow, anim
0
$VAR_STR 2 PGMIX 1 ASCII rod4 sezgenot gemat 0.20      #2 = second random effect,
        # PGMIX= type of combined relationship, 1 = sire+dam, rod4= name of pedigree,
        # sezgenot= name of file of genotyped animals, gemat= file with genomic relationship,
        # 0.20= weight of pedigree relationship
$PRIOR
1 1 1  6.8489355       # 1 = triangle of permanent environment covariance matrix (12x12) for
1 2 1  0.3630769       #first random effect cow
1 2 2  1.5650312
1 3 1 -0.075673
1 3 2  0.1232025
1 3 3  0.5213394
1 4 1 -0.061666
1 4 2 -0.071714
1 4 3  0.029643
1 4 4  0.2435163
1 5 1  3.7511534
1 5 2  0.7428758
1 5 3  0.0136907
1 5 4 -0.002865
1 5 5 11.17252
1 6 1   0.1681511
1 6 2   0.2836088
1 6 3 -0.048012
1 6 4   0.0005802
```

1 6 5 0.6147304
1 6 6 2.7287618
1 7 1 -0.101992
1 7 2 -0.092613
1 7 3 0.0438634
1 7 4 0.0110097
1 7 5 -0.457097
1 7 6 0.0952177
1 7 7 0.9689733
1 8 1 -0.05431
1 8 2 -0.002055
1 8 3 0.0091667
1 8 4 0.0179287
1 8 5 -0.101704
1 8 6 -0.191659
1 8 7 -0.026758
1 8 8 0.3700977
1 9 1 3.2634698
1 9 2 0.4441378
1 9 3 0.0086804
1 9 4 -0.094197
1 9 5 6.1256801
1 9 6 1.0718576
1 9 7 -0.183431
1 9 8 -0.146208
1 9 9 12.923124
1 10 1 0.2150463
1 10 2 0.177847
1 10 3 -0.053467
1 10 4 0.0074918
1 10 5 0.1941889
1 10 6 0.7177153
1 10 7 0.0662482
1 10 8 -0.141721
1 10 9 0.4005584
1 10 10 3.0229527
1 11 1 0.0275057
1 11 2 -0.01371
1 11 3 0.0151852
1 11 4 0.0096294
1 11 5 -0.211333
1 11 6 -0.028853
1 11 7 0.2556847
1 11 8 -0.010958
1 11 9 -0.489477
1 11 10 0.1495489
1 11 11 1.1552266
1 12 1 -0.081858
1 12 2 -0.062981
1 12 3 0.0019495
1 12 4 0.0083403
1 12 5 -0.067442

```
1 12 6    -0.181254
1 12 7    -0.022478
1 12 8     0.0618851
1 12 9    -0.059229
1 12 10  -0.214024
1 12 11  -0.076793
1 12 12   0.399678
2 1 1  3.3896411        # 2 = triangle of genetic covariance matrix (12x12) for second random
2 2 1  0.3046061        #effect animal
2 2 2  0.4755081
2 3 1 -0.479661
2 3 2  0.0248085
2 3 3  0.3157532
2 4 1  0.176901
2 4 2  0.0016134
2 4 3 -0.104984
2 4 4  0.0739097
2 5 1  3.60412
2 5 2  0.4816699
2 5 3 -0.493112
2 5 4  0.1853526
2 5 5  4.8106244
2 6 1 -0.093063
2 6 2  0.6455554
2 6 3  0.3191798
2 6 4 -0.11972
2 6 5  0.199605
2 6 6  1.7407905
2 7 1 -0.320409
2 7 2  0.0783388
2 7 3  0.2878517
2 7 4 -0.080907
2 7 5 -0.340016
2 7 6  0.3766198
2 7 7  0.3786789
2 8 1  0.1183183
2 8 2  0.009996
2 8 3 -0.092611
2 8 4  0.0574635
2 8 5  0.112859
2 8 6 -0.112999
2 8 7 -0.07372
2 8 8   0.0959358
2 9 1   3.4580753
2 9 2   0.386692
2 9 3  -0.515037
2 9 4   0.1885794
2 9 5   4.5717311
2 9 6   0.201976
2 9 7  -0.404006
2 9 8   0.1031577
2 9 9   5.3210489
```

```
2 10 1  0.1170898
2 10 2  0.7235655
2 10 3  0.3137405
2 10 4 -0.094165
2 10 5  0.4481552
2 10 6  1.7462697
2 10 7  0.415393
2 10 8 -0.096003
2 10 9  0.3716594
2 10 10 2.1440625
2 11 1   -0.486612
2 11 2    0.0386035
2 11 3    0.2798391
2 11 4   -0.086951
2 11 5   -0.482291
2 11 6    0.409802
2 11 7    0.3209385
2 11 8   -0.089642
2 11 9   -0.504233
2 11 10  0.4336006
2 11 11  0.5252568
2 12 1    0.1616494
2 12 2   -0.016537
2 12 3   -0.110833
2 12 4    0.0504514
2 12 5    0.1605075
2 12 6   -0.150833
2 12 7   -0.129177
2 12 8    0.0713992
2 12 9    0.1570471
2 12 10 -0.126288
2 12 11 -0.152298
2 12 12  0.1654504
3 1 1  4.58820            # 3 = triangle of (3 x 3) residual covariance matrix
3 2 1 0
3 2 2 7.36
3 3 1 0
3 3 2 0
3 3 3 8.57
$DMU5
30000 0.1E-11
```

III. Novelty of approaches

Presented notebook is combination of theory of linear models with algorithms of practical calculations by own programming and using of available software. Presented examples cover different situations, which can users meet in practise. Examples can be easily modified and used like guide for constructions of own parameter files for practical calculation. Presented methodology is a new in a field of application of linear models and in a genetic evaluation.

IV. Description of application

Users of methodology are principally persons working in nation-wide evaluation of animals (in Czech-Moravian Corporation of Animal Breeders), but could be used also by scientists of different professions and used for education of students at universities.

V. Economic standpoints

Methodology serves for nation-wide evaluation of animals, which is by a law No. 110/1997 Sb. and by a law No. 154/2000 Sb. of The Czech Republic, run by authorised organization (Czech-Moravian Corporation of Animal Breeders). Therefore it serves for national information system and state administrative. Results are published in favour of all breeders in a country. Czech-Moravian Corporation of Animal Breeders and association of breeders, which intermediate results to breeders were established by a law like not-profit organization. Potential profit from application of new procedures will be generated by all farmers in a country in their agriculture production.

VI. References

Christensen O. F. and Lund M. S. (2010): Genomic prediction when some animals are not genotyped. Genet. Sel. Evol., 42:2.

Forni S., Aguilar I., Misztal I. (2011): Different genomic relationship matrices for single-step analysis using phenotypic, pedigree and genomic information. Genet. Sel. Evol., 43, 1.

Henderson C.R. (1976): Simple method for computing inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics, 31, 69-83.

Jairath L., Dekkers J.C.M., Schaeffer L.R., Liu Z., Burnside E.B., Kolstad B. (1998): Genetic evaluation for herd life in Canada. J. Dairy Sci., 81, 550-562.

Legarra A., Christensen O.F., Aguilar I., Misztal I. (2014): Single step, a general approach for genomic selection. Livestock Sci., 166, 54-65.

Legara A., Misztal I. (2008): Technical note: Computing Strategies in Genome-Wide Selection. J. Dairy Sci., 91, 360-366.

Lidauer M., Strandén I., Mäntysaari E.A., Pösö J., Kettunen A. (1999): Solving Large Test-Day Models by Iteration on Data and Preconditioned Conjugate Gradient. J. Dairy Sci., 82, 2788-2796.

Madsen P., Jensen J. (2010): A user guide to DMU, version 6, release 5.0. Manual, Faculty of Agricultural Science, University of Aarhus. Retrieved on from *http://dmu.agrsci.dk.*

Madsen P., Su G., Labouriau R., Christensen O.F. (2010): DMU - A package for analysing multivariate mixed models. 9th World Congr. Genet. Appl. Livest. Prod. (WCGALP), Leipzig, Germany (0732).

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

Misztal I. Legarra A, Aguilar I. (2009): Computing procedures for genetic evaluation including phenotypic, full pedigree, and genomic information. J. Dairy Sci., 92: 4648-4655.

Misztal I, Tsuruta S, Strabel T, Auvray B, Druet T. Lee D.H. (2002): BLUPF90 and related programs (BGF90). In the 7th World Congr. Genet. Appl. Livest. Prod. (WCGALP), pp. 28, Montpellier, France. Retrieved on from *http://nce.ads.uga.edu/wiki/doku.php.*

Mrode R. (2014): Linear models for the prediction of animal breeding values. 3rd edition. ISBN 9781780643915.

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

SAS (2014): Statistical analysis system. http://www.sas.com.
Schaeffer L.R. (2014): Lectures. http://www.aps.uoguelph.ca/~lrs/ABModels/notesx.html.
Schaeffer L.R. (1994): Multiple-country comparison of dairy sires. J. Dairy Sci., 77, 2671-2678.
Su G. Madsen P. (2011): User's Guide for Gmatrix. A program for computing Genomic relationship matrix. Retrieved on from http://dmu.agrsci.dk.
Szyda J. Żarnecki, A., Suchocki, T. (2011): Fitting and validating the genomic evaluation model to Polish Holstein-Friesian cattle. J. Appl. Genet., 52, 363-366.
Tsuruta S., Misztal I., Strandén I. (2001): Use of preconditioned conjugate gradient algorithm as a generic solver for mixed-model equations in animal breeding applications. J. Anim. Sci., 79, 1166-1172.
VanRaden P. M. (2008): Efficient methods to compute genomic predictions. J. Dairy Sci., 91, 4414–4423.
Vitezica Z.G., Aguilar I., Misztal I., Legarra A. (2011): Bias in genomic predictions for populations under selection. Genet. Res., 93, 357–366.

VII. Own publications preceding this methodology

Bauer, J., Milerski, M., Přibyl, J., Vostry, L. (2012): Estimation of genetic parameters and evaluation of test-day milk production in sheep. Czech J. Anim. Sci., 57, 522-528.
Pešek, J., Přibyl, J., Vostrý, L. (2014): Analysis of dairy cattle loci in relation to milk yield. International Conference XXVI Genetic Days, Praha, Czech Rep., 3-4 September. Book of abstracts, p. 130.
Přibyl J., Bauer J., Pešek P., Přibylová J., Vostrá Vydrová H., Vostrý L.,Zavadilová L. (2014): Domestic and Interbull information in the single step genomic evaluation of Holstein milk production. Czech J. Anim. Sci., 59, 409-415.
Přibyl J., Haman J., Kott T., Přibylová J., Šimečková M., Vostrý L., Zavadilová L., Čermák V., Růžička Z., Šplíchal J., Verner M., Motyčka J., Vondrášek L. (2012): Single-step prediction of genomic breeding value in a small dairy cattle population with strong import of foreign genes. Czech J. Anim. Sci., 57, 151-159.
Přibyl J., Madsen P., Bauer J., Přibylová J., Šimečková M., Vostrý L., Zavadilová L. (2013): Contribution of domestic production records, Interbull estimated breeding values, and single nucleotide polymorphism genetic markers to the single-step genomic evaluation of milk production. J. Dairy Sci., 96: 1865-1873.
Přibyl, J., Misztal, I., Přibylová, J., Šeba, K. (2003): Multiple-breed, Multiple-traits evaluation of beef cattle in the Czech Republic. Czech J. Anim. Sci. 48: 519-532.
Přibyl, J., Přibylová, J. (2002): Výběr vhodného modelu při vyhodnocování souboru údajů. XV. letní škola biometriky „Biometrické metody a modely v současné vědě a výzkumu". Lednice na Moravě, 2.- 6.9. Sborník referátů : 41-50. ÚKZÚZ v Brně.
Přibyl J., Řehout V., Čítek J., Přibylová J. (2010): Genetic evaluation of dairy cattle using a simple heritable genetic ground. J. Sci. Food Agric. 90:1765-1773.
Veselá, Z., Přibyl, J., Šafus P., Vostrý, L., Šeba, K., Štolc, L. (2005): Breeding value for type traits in beef cattle in the Czech Republic. Czech J. Anim. Sci. 50: 385-393.
Vostrý, L., Veselá, Z., Přibyl, J. (2012): Genetic parameters for growth of young beef bulls Arch. Tierzucht. 55: 245-254.
Zavadilová L, Jamrozik J., Schaeffer L.R. (2005a):. Genetic parameters for test-day model with random regressions for production traits of Czech Holstein cattle. Czech J. Anim. Sci., 50, 142-154.
Zavadilová L, Němcova E, Přibyl J., Wolf J. (2005b): Definition of subgroups for fixed regression in the test-day animal model for milk production of Holstein cattle in the Czech Republic. Czech J. Anim. Sci., 50, 7-13.

Publisher: Institute of animal science
Přátelství 815, 104 00 Praha Uhříněves, Czech Republic

Title: Genetic evaluation by Linear Models using own algorithms and standard software

Authors: J. Přibyl (proportion 15%), J. Bauer (10%), E. Krupa (10%), Z. Krupová (5%), M. Milerski (5%), A. Novotná (5%), P. Pešek (10%), J. Přibylová (5%), J. Schmidová (5%), A. Svitáková (5%), Z. Veselá (5%), H. Vostrá Vydrová (5%), L. Vostrý (5%), L. Zavadilová (5%), E. Žáková (5%)

ISBN: **978-80-7403-128-1**

Acknowledgements:
Elaborated with support by the Czech Ministry of Agriculture, Project QI111A167 (Genomic selection of dairy cattle).