

Supplement

Supplement 1

Overview and general description of toy examples

Six toy examples for generalized gametic imprinting models

Supplement 2

Example pedigree

Lower triangular matrix (decomposed inverse generalized gametic relationships)

Supplement 3

Detailed description of mouse data analyses and results

Supplement 1

Overview and general description of toy examples

Six toy examples for generalized gametic imprinting models

- 1) The gametic model
- 2) The reduced gametic model
- 3) The generalized gametic model
- 4) The reduced generalized gametic model
- 5) The gametic model with maternal genetic effect
- 6) The generalized gametic model with maternal gametic effect

Description of toy examples

Inga Blunk and Norbert Reinsch

30 Sep 2020

To investigate the effects of imprinted loci, gametic relationship matrices (e.g. Schaeffer *et al.*, 1989) have been used in pedigree-based imprinting analyses using different kinds of *gametic models*. One drawback of this is the size of the matrices because they represent each individual by two gametic effects. In our research article “*Generalized gametic relationships for flexible analyses of parent-of-origin effects*”, we addressed this issue by proposing a combination of average gametic effects (transmitting abilities; TAs) for individuals without records and single gametic effects for others. In the following the corresponding statistical model is referred to as *generalized gametic model*. The model uses a generalized gametic relationship matrix $\bar{\mathbf{G}}$ that represents the covariance of the mixture of TAs and gametic effects.

Examples of *gametic model* versions To illustrate the development of the *generalized gametic model* six toy examples are provided each demonstrating a different version of *gametic model*:

1. *gametic model*: utilizes the gametic relationship matrix $\mathbf{G}$ (Schaeffer *et al.*, 1989) and predicts two gametic effects for all individuals in the pedigree (including final progeny). The example is stored and can be navigated from the directory *1_GAM*.
2. *reduced gametic model*: reduces the number of equations by excluding the final progeny. It utilizes the gametic relationship matrix $\mathbf{G}$ that now represents the covariance between the gametic effects of parents, i.e. gametic effects are predicted for parents only. The example is stored and can be navigated from the directory *2_GamRed*.
3. *generalized gametic model*: utilizes the generalized gametic relationship matrix $\bar{\mathbf{G}}$, i.e. predicts two gametic effects for individuals with records and TAs for individuals without records. In comparison to the *gametic model* the *generalized gametic model* saves one equation for all un-phenotyped parents because only the average gametic effect (TA) is predicted. The example is stored and can be navigated from the directory *3_GenGam*.
4. *reduced generalized gametic model*: further reduces the number of equations in comparison to the *generalized gametic model* because final progeny are excluded. The model still allows for the inclusion of parents with records. The example is stored and can be navigated from the directory *4_RedGenGam*.
5. *gametic model with maternal genetic effects*: corresponds to the *gametic model* in (1) but includes a maternal genetic effect in order to separate “maternal imprinting effects” from “maternal *non*-imprinting effects”. As the model predicts two gametic effects and a maternal genetic effect for all individuals in the pedigree, this model generates the largest system of equations. The example is stored and can be navigated from the directory *5_GamMat*.
6. *generalized gametic model with maternal genetic effects*: corresponds to the *generalized gametic model* in (3) but includes a maternal genetic effect. The covariance of this effect is assumed to be $\bar{\mathbf{G}}$. The example is stored and can be navigated from the directory *6_GenGamMat*.

Schaeffer *et al.*, 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Toy example - the gametic model

Inga Blunk

1 Apr 2020

This script provides an example for the *gametic model*. In matrix notation the model can be written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{gs}\mathbf{g}_s + \mathbf{Z}_{gd}\mathbf{g}_d + \mathbf{e},$$

where $\mathbf{g}_s$ and $\mathbf{g}_d$ are vectors containing gametic effects under a paternal expression pattern and gametic effects under a maternal expression pattern. This means that for all animals in a pedigree (for parents with and without records and for final progeny with records) two parental breeding values are estimated which constitute compositions of both parental gametic effects, respectively. The matrices $\mathbf{Z}_{gs}$ and $\mathbf{Z}_{gd}$ connect the observations with the expressed gametes. The gametic variances are:

$$\text{Var} \begin{bmatrix} \mathbf{g}_s \\ \mathbf{g}_d \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{G}\sigma_s^2 & \mathbf{G}\sigma_{sd} & 0 \\ \mathbf{G}\sigma_{sd} & \mathbf{G}\sigma_d^2 & 0 \\ 0 & 0 & \mathbf{I}\sigma_e^2 \end{bmatrix}.$$

$\mathbf{G}$ is the gametic relationship matrix (Schaeffer *et al.*, 1989). The gametic variances are used in the mixed model equations as:

$$\begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_2 & \lambda_3 \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & \sigma_{sd} \\ \sigma_{sd} & \sigma_d^2 \end{bmatrix}^{-1} \sigma_e^2.$$

The mixed model equations are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z}_{gs} & \mathbf{X}'\mathbf{Z}_{gd} \\ \mathbf{Z}_{gs}'\mathbf{X} & \mathbf{Z}_{gs}'\mathbf{Z}_{gs} + \mathbf{G}^{-1}\lambda_1 & \mathbf{Z}_{gs}'\mathbf{Z}_{gd} + \mathbf{G}^{-1}\lambda_2 \\ \mathbf{Z}_{gd}'\mathbf{X} & \mathbf{Z}_{gd}'\mathbf{Z}_{gs} + \mathbf{G}^{-1}\lambda_2 & \mathbf{Z}_{gd}'\mathbf{Z}_{gd} + \mathbf{G}^{-1}\lambda_3 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{g}}_s \\ \hat{\mathbf{g}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}_{gs}'\mathbf{y} \\ \mathbf{Z}_{gd}'\mathbf{y} \end{bmatrix}.$$

Solving the mixed model equations by direct inversion of the coefficient matrix, provides $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{g}}_s$, and $\hat{\mathbf{g}}_d$.

References

- Gilmour, A.R., Gogel, B.J., Cullis, B.R., and Thompson R. 2009. ASReml user guide release 3.0. VSN International Ltd., Hemel Hempstead, UK. <https://asreml.kb.vsnr.co.uk/wp-content/uploads/sites/3/2018/02/ASReml-3-User-Guide.pdf>
- Schaeffer, L.R., Kennedy, B.W., and Gibson, J.P. 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Example

In the following a toy example is provided in an R environment, i.e. based on a toy dataset the mixed model equations of the *gametic model* will be solved in R. Subsequently, the same dataset will be analyzed with ASReml to show how the *gametic model* can be implemented into the software. Eventually, the R and ASReml solutions will be compared. At the end of the R script breeding values and imprinting effects will be calculated. They can be compared to the results generated with the other *gametic models* in order to show their equivalences. The fortran program *GGRinv* that builds the inverse gametic relationship matrix $\mathbf{G}$ and the ASReml software (version 3.0; Gilmour *et al.*, 2009) can be navigated from the following R script. The corresponding command files (command_file.txt for *GGRinv*; command_file.as for ASReml) must be available within the corresponding directories, which are *ProFor* and *ASReml*, respectively. The example was build in an UNIX environment.

```

### Overall number of animals in the pedigree:

nr_animals = 14

### data file:

# F = inbreeding coefficient; y = phenotype

data.ped = data.frame(id = seq(1,nr_animals,1),
                      dad = c(0,0,1,1,3,3,1,1,3,5,10,5,10,5),
                      mom = c(0,0,0,0,4,4,4,4,2,2,9,8,9,8),
                      F = c(0,0,0,0,0.125,0.125,0.250,0.250,0,0,0.203125,0.28125,0.203125,0.28125),
                      y = c(NA,NA,NA,NA,NA,444,555,550,NA,580,625,375,400,355))

data.ped # toy data to be analyzed in the following:

##      id dad mom      F      y
## 1     1   0   0 0.000000    NA
## 2     2   0   0 0.000000    NA
## 3     3   1   0 0.000000    NA
## 4     4   1   0 0.000000    NA
## 5     5   3   4 0.125000    NA
## 6     6   3   4 0.125000  444
## 7     7   1   4 0.250000  555
## 8     8   1   4 0.250000  550
## 9     9   3   2 0.000000    NA
## 10    10   5   2 0.000000  580
## 11    11  10   9 0.203125  625
## 12    12   5   8 0.281250  375
## 13    13  10   9 0.203125  400
## 14    14   5   8 0.281250  355

#
### gametic variance components:
sigma.s.2 = 2552      # gametic variance as father
sigma.d.2 = 2800      # gametic variances as mother
sigma.sd = 2670       # gametic covariance
sigma.e.2 = 12756     # residual variance in a gametic model

### construct lambdas for the coefficient matrix (left hand side of mixed model equations):
det.Var = sigma.s.2*sigma.d.2-(sigma.sd*sigma.sd)

la1 = (1/det.Var)*sigma.d.2*sigma.e.2
la2 = (1/det.Var)*(-sigma.sd)*sigma.e.2
la3 = (1/det.Var)*sigma.s.2*sigma.e.2

### construction of the inverse of the gametic relationship matrix with the fortran
### program "GGRInv" (as 3-column lower triangle):
setwd("ProFor") # change to the ProFor directory containing "GGRInv"

# prepare the dataset needed for "GGRInv". The dataset must provide the following
# columns: "id", "dad", "mom", "indicator", "F"
data_in = cbind(data.ped[,1:3],rep(2,nrow(data.ped)),data.ped$F)
colnames(data_in)[c(4,5)] = c("indicator", "F")

```

```
data_in # note: indicator vector of 2s -> construction of full gametic relationship matrix G
```

```
##      id dad mom indicator      F
## 1    1   0   0          2 0.000000
## 2    2   0   0          2 0.000000
## 3    3   1   0          2 0.000000
## 4    4   1   0          2 0.000000
## 5    5   3   4          2 0.125000
## 6    6   3   4          2 0.125000
## 7    7   1   4          2 0.250000
## 8    8   1   4          2 0.250000
## 9    9   3   2          2 0.000000
## 10  10   5   2          2 0.000000
## 11  11  10   9          2 0.203125
## 12  12   5   8          2 0.281250
## 13  13  10   9          2 0.203125
## 14  14   5   8          2 0.281250
```

```
# write dataset to ProFor directory:
```

```
write.table(data_in, "pedigree_input.txt", col.names=FALSE, row.names=FALSE, quote=FALSE, sep="\t")
```

```
# run "GGRInv" to construct the inverse of G (AImatrix.giv) with system function:
```

```
#####
#
# system("GGRInv > out.txt") #
#
#####
```

```
# move inverse of G (AImatrix.giv) to the ASreml directory so it will be available for the
# subsequent analysis with the ASreml software:
```

```
# system("mv AImatrix.giv ../ASreml")
```

```
### rewrite inverse gametic relationship matrix to a n x n matrix:
```

```
setwd("../ASreml") # change to the ASreml directory
```

```
GAMinv = read.table("AImatrix.giv", header=F) # 3-column lower triangle of inverse of G
```

```
GAMi = matrix(ncol=nr_animals*2, nrow=nr_animals*2, 0) # inverse of G as n x n matrix
```

```
i = 0
```

```
for(i in 1:nrow(GAMinv)){
```

```
  GAMi[ GAMinv[i,1] , GAMinv[i,2]] = GAMinv[i,3]
```

```
  GAMi[ GAMinv[i,2], GAMinv[i,1]] = GAMinv[i,3]
```

```
}
```

```
GAMi[1:9,1:10]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]      [,9]      [,10]
## [1,]    3    2    0    0 -1.0  0.0  -1    0  0.000000  0.000000
## [2,]    2    3    0    0 -1.0  0.0  -1    0  0.000000  0.000000
## [3,]    0    0    2    1  0.0  0.0    0    0  0.000000  0.000000
## [4,]    0    0    1    2  0.0  0.0    0    0  0.000000  0.000000
## [5,]   -1   -1    0    0  3.5  1.5    0    0 -1.000000  0.000000
## [6,]    0    0    0    0  1.5  2.5    0    0 -1.000000  0.000000
## [7,]   -1   -1    0    0  0.0  0.0    4    2  0.000000 -1.000000
## [8,]    0    0    0    0  0.0  0.0    2    3  0.000000 -1.000000
## [9,]    0    0    0    0 -1.0 -1.0    0    0  3.714286  1.714286
```

```

### construction of incidence matrices to set up the mixed model equations:
beob = data.ped[is.na(data.ped$y) == FALSE,1] # IDs of animals with phenotypes

## Z.gs
id.dad = data.ped$dad
id.dad=unique(id.dad[-(which(id.dad == 0))])
id.mom = data.ped$mom
id.mom = unique(id.mom[-(which(id.mom == 0))])
index_p = matrix(nrow = nr_animals, ncol = 2,0); index_p[,1] = seq(1,nr_animals); l = 0; la = 0

for(k in 1:nr_animals){
  if(k == 1){l = 1; la = 1}else{l = k+la ; la = la+1}
  index_p[k,2] = 1
}
Z.gs = matrix(ncol = (nr_animals)*2, nrow = length(beob),0)
for(i in 1:length(which(is.na(data.ped$y)==FALSE))){
  tier=beob[i]
  ii = index_p[index_p[,1] == tier,2]
  Z.gs[i,ii] = 1
}

Z.gs # note that the records of animals are linked to their expressed paternal gametes:

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    0    0    0    0    0    1    0    0
## [2,]    0    0    0    0    0    0    0    0    0    0    0    0    1
## [3,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [4,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,]    0    0    0    0    0    0    0    0    0    0    0
## [2,]    0    0    0    0    0    0    0    0    0    0    0
## [3,]    0    1    0    0    0    0    0    0    0    0    0
## [4,]    0    0    0    0    0    1    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    1    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    1    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0
##      [,25] [,26] [,27] [,28]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
## [4,]    0    0    0    0
## [5,]    0    0    0    0
## [6,]    0    0    0    0
## [7,]    1    0    0    0
## [8,]    0    0    1    0

## Z.gd
index_m = matrix(nrow = nr_animals, ncol = 2, 0); index_m[,1] = seq(1,nr_animals); l = 0; la = 0

```

```

for(k in 1:nr_animals){
  if(k == 1){l = 2; la = 2}else{l = k+la ; la = la+1}
  index_m[k,2] = l
}
Z.gd = matrix(ncol = (nr_animals)*2, nrow = length(beob),0)
for(i in 1:length(beob)){
  tier = beob[i]
  ii = index_m[index_m[,1] == tier,2]
  Z.gd[i,ii] = 1
}
Z.gd # note that the records of animals are linked to their expressed maternal gametes:

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    0    0    0    0    0    0    1    0
## [2,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [3,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [4,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,]      0      0      0      0      0      0      0      0      0      0      0
## [2,]      1      0      0      0      0      0      0      0      0      0      0
## [3,]      0      0      1      0      0      0      0      0      0      0      0
## [4,]      0      0      0      0      0      0      1      0      0      0      0
## [5,]      0      0      0      0      0      0      0      0      1      0      0
## [6,]      0      0      0      0      0      0      0      0      0      0      1
## [7,]      0      0      0      0      0      0      0      0      0      0      0
## [8,]      0      0      0      0      0      0      0      0      0      0      0
##      [,25] [,26] [,27] [,28]
## [1,]      0      0      0      0
## [2,]      0      0      0      0
## [3,]      0      0      0      0
## [4,]      0      0      0      0
## [5,]      0      0      0      0
## [6,]      0      0      0      0
## [7,]      0      1      0      0
## [8,]      0      0      0      1

```

```

## X: the only fixed effect is the population mean:
X = matrix(ncol = 1,nrow = length(beob),1)
X

```

```

##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
## [4,]    1
## [5,]    1
## [6,]    1
## [7,]    1
## [8,]    1

```

```

## y
y = as.matrix(data.ped[which(is.na(data.ped$y) == FALSE),which(colnames(data.ped) == "y")])
y

##      [,1]
## [1,] 444
## [2,] 555
## [3,] 550
## [4,] 580
## [5,] 625
## [6,] 375
## [7,] 400
## [8,] 355

### set up the mixed model equations:
LHS = rbind(cbind((t(X)%*%X), # left hand side (coefficient matrix)
                  (t(X)%*%Z.gs),
                  (t(X)%*%Z.gd)),
            cbind((t(Z.gs)%*%X),
                  (t(Z.gs)%*%Z.gs+GAMi*la1),
                  (t(Z.gs)%*%Z.gd+GAMi*la2)),
            cbind((t(Z.gd)%*%X),
                  (t(Z.gd)%*%Z.gs+GAMi*la2),
                  (t(Z.gd)%*%Z.gd+GAMi*la3)))

RHS = rbind((t(X)%*%y),t(Z.gs)%*%y,t(Z.gd)%*%y) # right hand side

### Solving the mixed model equations by direct inversion of the coefficient matrix
### -> solutions in LS
LS = solve(LHS)%*%RHS

#####
##### ASREML: #####
#####

# Now, the results generated in R (saved in vector LS) should be replicated with ASreml.
# For this purpose, the dataset needed for ASreml can be prepared in R.

# The dataset must contain the columns ID = individual; g1nr = paternal gamete; g2nr =
# maternal gamete; and the phenotype:
ID = data.ped[,1]
g1nr = (ID-1)*2 + 1
g2nr = g1nr + 1
data.ped[is.na(data.ped$y)==TRUE,colnames(data.ped)=="y"] = 0
PP = as.data.frame(cbind(ID, g1nr, g2nr, data.ped$y)); colnames(PP)[4]="phenotype"
PP

```

##	ID	g1nr	g2nr	phenotype
## 1	1	1	2	0
## 2	2	3	4	0
## 3	3	5	6	0
## 4	4	7	8	0
## 5	5	9	10	0
## 6	6	11	12	444
## 7	7	13	14	555

```

## 8 8 15 16 550
## 9 9 17 18 0
## 10 10 19 20 580
## 11 11 21 22 625
## 12 12 23 24 375
## 13 13 25 26 400
## 14 14 27 28 355

# write dataset (PP) to ASreml directory:
write.table(PP, "dataset.txt", col.names=TRUE, row.names=FALSE, quote=F, sep="\t")

### run ASreml from R-script using the system function:

#####
#
# system("asreml -l command_file.as") #
#
#####

### Now, compare results of R and ASreml:
ASR = read.table("command_file.sln", header=FALSE) # read in the ASreml solution file:

results = as.data.frame(cbind(ASR[,3],round(LS,3)))
colnames(results)=c("ASreml_solution","R_solution")
results

## ASreml_solution R_solution
## 1 493.4000 493.440
## 2 -3.1440 -3.144
## 3 -3.1440 -3.144
## 4 9.3270 9.327
## 5 9.3270 9.327
## 6 -6.8850 -6.885
## 7 -7.4820 -7.482
## 8 -5.5860 -5.586
## 9 -4.8840 -4.884
## 10 -12.5200 -12.520
## 11 -10.5700 -10.572
## 12 -10.2200 -10.224
## 13 -8.4160 -8.416
## 14 2.6590 2.659
## 15 0.8369 0.837
## 16 -5.9100 -5.910
## 17 -7.6750 -7.675
## 18 -6.2890 -6.289
## 19 10.2200 10.222
## 20 -4.4600 -4.460
## 21 17.7600 17.760
## 22 16.8100 16.807
## 23 12.5900 12.594
## 24 -19.0100 -19.010
## 25 -13.4900 -13.485
## 26 -1.7970 -1.797
## 27 -6.8700 -6.870
## 28 -20.5100 -20.506

```

```
## 29      -14.8300    -14.827
## 30      -3.3120     -3.312
## 31      -3.3120     -3.312
## 32       9.7790     9.779
## 33       9.7790     9.779
## 34      -7.2250     -7.225
## 35      -7.8250     -7.825
## 36      -5.8670     -5.867
## 37      -5.1090     -5.109
## 38     -13.1100    -13.108
## 39     -11.0700    -11.071
## 40     -10.7100    -10.706
## 41      -8.8230     -8.823
## 42       2.7600     2.760
## 43       0.8801     0.880
## 44      -6.2290     -6.229
## 45      -8.0460     -8.046
## 46      -6.5870     -6.587
## 47      10.7200     10.718
## 48      -4.6770     -4.677
## 49      18.6200     18.620
## 50      17.6000     17.599
## 51      13.2100     13.210
## 52     -19.9000    -19.898
## 53     -14.1600    -14.157
## 54      -1.8650     -1.865
## 55      -7.2020     -7.202
## 56     -21.4600    -21.464
## 57     -15.5600    -15.564
```

```
#####
```

```
# use the solutions to calculate the breeding values and imprinting effects:
```

```
LSi=LS[-1]
```

```
LSasFather=LSi[1:(nr_animals*2)]
```

```
LSasMother=LSi[(nr_animals*2+1):(nr_animals*4)]
```

```
LSasFathergv=LSasFather[seq(1,(nr_animals*2),2)]
```

```
LSasFathergm=LSasFather[seq(2,(nr_animals*2),2)]
```

```
asV=cbind(LSasFathergv,LSasFathergm)
```

```
asV # gametic effects as sire:
```

```
##      LSasFathergv LSasFathergm
## [1,]    -3.144411    -3.144411
## [2,]     9.327354     9.327354
## [3,]    -6.885258    -7.481695
## [4,]    -5.586481    -4.884141
## [5,]   -12.519811   -10.571646
## [6,]   -10.223727    -8.416138
## [7,]     2.659439     0.836898
## [8,]    -5.909752    -7.674500
## [9,]    -6.288586    10.222245
## [10,]   -4.460332    17.759818
## [11,]    16.807183    12.593932
```

```
## [12,] -19.009526 -13.485474
## [13,] -1.797016 -6.870493
## [14,] -20.505924 -14.827407

LSasMothergv=LSasMother[seq(1,(nr_animals*2),2)]
LSasMothergm=LSasMother[seq(2,(nr_animals*2),2)]
asM=cbind(LSasMothergv,LSasMothergm)
asM # gametic effects as dam:
```

```
##      LSasMothergv LSasMothergm
## [1,] -3.312217 -3.3122173
## [2,]  9.779399  9.7793988
## [3,] -7.224938 -7.8254422
## [4,] -5.866651 -5.1088670
## [5,] -13.108268 -11.0708364
## [6,] -10.706017 -8.8234571
## [7,]  2.759992  0.8801009
## [8,] -6.229489 -8.0457101
## [9,] -6.586728 10.7178606
## [10,] -4.676539 18.6203358
## [11,] 17.599001 13.2100932
## [12,] -19.898463 -14.1568406
## [13,] -1.865424 -7.2020374
## [14,] -21.464052 -15.5641113
```

```
## breeding values and imprinting effects:
BVasS=asV[,1]+asV[,2] # breeding value as sire
BVasM=asM[,1]+asM[,2] # breeding value as dam
EIE = BVasM - BVasS # imprinting effects
```

```
cbind(BVasS,BVasM,EIE)
```

```
##      BVasS      BVasM      EIE
## [1,] -6.288821 -6.624435 -0.3356135
## [2,] 18.654709 19.558798  0.9040890
## [3,] -14.366953 -15.050381 -0.6834275
## [4,] -10.470622 -10.975518 -0.5048955
## [5,] -23.091457 -24.179104 -1.0876470
## [6,] -18.639865 -19.529474 -0.8896091
## [7,]  3.496337  3.640093  0.1437559
## [8,] -13.584253 -14.275199 -0.6909466
## [9,]  3.933659  4.131132  0.1974736
## [10,] 13.299486 13.943796  0.6443103
## [11,] 29.401115 30.809094  1.4079787
## [12,] -32.495000 -34.055303 -1.5603030
## [13,] -8.667508 -9.067461 -0.3999529
## [14,] -35.333332 -37.028163 -1.6948314
```

```
setwd("../")
```

Toy example - the reduced gametic model

Inga Blunk

1 Apr 2020

This script provides an example for the *reduced gametic model*. In matrix notation the model can be written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{rs}\mathbf{g}_s + \mathbf{Z}_{rd}\mathbf{g}_d + \boldsymbol{\epsilon},$$

where $\mathbf{g}_s$ and $\mathbf{g}_d$ are vectors containing gametic effects under a paternal expression pattern and gametic effects under a maternal expression pattern *for parents only*. The matrices $\mathbf{Z}_{rs}$ and $\mathbf{Z}_{rd}$ connect observations of final progeny with the gametic effects of their parents (entry of 0.5 at the position of the parental gametes) and observations of parents with their own gametic effects (entry of 1 at the position of the gametes of a phenotyped animal). The vector $\boldsymbol{\epsilon}$ is a vector of residuals, which either is $\mathbf{r}$ for records from final progeny that are linked to the genetic effects of their parents or $\mathbf{e}$ for records from parents represented by their two gametic effects. The gametic variances are:

$$\text{Var} \begin{bmatrix} \mathbf{g}_s \\ \mathbf{g}_d \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_r\sigma_s^2 & \mathbf{G}_r\sigma_{sd} & 0 \\ \mathbf{G}_r\sigma_{sd} & \mathbf{G}_r\sigma_d^2 & 0 \\ 0 & 0 & \mathbf{W}\sigma_e^2 \end{bmatrix}.$$

$\mathbf{G}_r$ is the gametic relationship matrix (Schaeffer *et al.*, 1989) that now only contains the gametic relationships between parents and their ancestors, while matrix $\mathbf{W}$ is a diagonal matrix containing weightings of observations. For observations from parents the diagonal elements are $w = 1$. For observations from final progeny w is:

$$\frac{0.5\sigma_s^2(1 - F_{si}) + 0.5\sigma_d^2(1 - F_{di}) + \sigma_e^2}{\sigma_e^2},$$

where F_{si} and F_{di} are the inbreeding coefficients of parents of phenotyped final progeny. The gametic variances are used in the mixed model equations as:

$$\begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_2 & \lambda_3 \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & \sigma_{sd} \\ \sigma_{sd} & \sigma_d^2 \end{bmatrix}^{-1} \sigma_e^2.$$

The mixed model equations are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{W}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{W}^{-1}\mathbf{Z}_{rs} & \mathbf{X}'\mathbf{W}^{-1}\mathbf{Z}_{rd} \\ \mathbf{Z}_{rs}'\mathbf{W}^{-1}\mathbf{X} & \mathbf{Z}_{rs}'\mathbf{W}^{-1}\mathbf{Z}_{rs} + \mathbf{G}_r^{-1}\lambda_1 & \mathbf{Z}_{rs}'\mathbf{W}^{-1}\mathbf{Z}_{rd} + \mathbf{G}_r^{-1}\lambda_2 \\ \mathbf{Z}_{rd}'\mathbf{W}^{-1}\mathbf{X} & \mathbf{Z}_{rd}'\mathbf{W}^{-1}\mathbf{Z}_{rs} + \mathbf{G}_r^{-1}\lambda_2 & \mathbf{Z}_{rd}'\mathbf{W}^{-1}\mathbf{Z}_{rd} + \mathbf{G}_r^{-1}\lambda_3 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{g}}_s \\ \hat{\mathbf{g}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{W}^{-1}\mathbf{y} \\ \mathbf{Z}_{rs}'\mathbf{W}^{-1}\mathbf{y} \\ \mathbf{Z}_{rd}'\mathbf{W}^{-1}\mathbf{y} \end{bmatrix}.$$

Solving the mixed model equations by direct inversion of the coefficient matrix, provides $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{g}}_s$ and $\hat{\mathbf{g}}_d$.

References

- Gilmour, A.R., Gogel, B.J., Cullis, B.R., and Thompson R. 2009. ASReml user guide release 3.0. VSN International Ltd., Hemel Hempstead, UK. <https://asreml.kb.vsnl.co.uk/wp-content/uploads/sites/3/2018/02/ASReml-3-User-Guide.pdf>
- Schaeffer, L.R., Kennedy, B.W., and Gibson, J.P. 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Example

In the following a toy example is provided in an R environment, i.e. based on a toy dataset the mixed model equations of the *reduced gametic model* will be solved in R. Subsequently, the same dataset will be

analyzed with ASreml to show how the model can be implemented into the software. Eventually, the R and ASreml solutions will be compared. At the end of the R script breeding values and imprinting effects will be calculated. They can be compared to the results generated with the other corresponding *gametic models* in order to show their equivalences. The fortran program *GGRinv* that builds the inverse gametic relationship matrix $\mathbf{G}$ and the ASreml software (version 3.0; Gilmour *et al.*, 2009) can be navigated from the following R script. The corresponding command files (command_file.txt for *GGRinv*; command_file.as for ASreml) must be available within the corresponding directories, which are *ProFor* and *ASreml*, respectively. The example was build in an UNIX environment.

```
### Overall number of animals in the pedigree:

nr_animals = 14

### data file:
# F = inbreeding coefficient; Fs = F of father; Fd = F of mother; y = phenotype
data.ped = data.frame(id = seq(1,nr_animals,1),
  dad = c(0,0,1,1,3,3,1,1,3,5,10,5,10,5),
  mom = c(0,0,0,0,4,4,4,4,2,2,9,8,9,8),
  F = c(0,0,0,0,0.125,0.125,0.250,0.250,0,0,0.203125,0.28125,0.203125,0.28125),
  y = c(NA,NA,NA,NA,NA,444,555,550,NA,580,625,375,400,355),
  Fs = c(0,0,0,0,0,0,0,0,0,0.125,0,0.125,0,0.125),
  Fd = c(0,0,0,0,0,0,0,0,0,0,0.25,0,0.25))

## who are the parents?
data.ped$parents = rep(0,nrow(data.ped))
# sires:
m=match(data.ped$id, data.ped$dad, nomatch=0)
data.ped[m!=0,colnames(data.ped)=="parents"]=1
# dams:
m=match(data.ped$id, data.ped$mom, nomatch=0)
data.ped[m!=0,colnames(data.ped)=="parents"]=1

data.ped # toy data to be analyzed in the following:

##      id dad mom      F    y    Fs    Fd parents
## 1      1  0  0 0.000000  NA 0.000 0.00      1
## 2      2  0  0 0.000000  NA 0.000 0.00      1
## 3      3  1  0 0.000000  NA 0.000 0.00      1
## 4      4  1  0 0.000000  NA 0.000 0.00      1
## 5      5  3  4 0.125000  NA 0.000 0.00      1
## 6      6  3  4 0.125000 444 0.000 0.00      0
## 7      7  1  4 0.250000 555 0.000 0.00      0
## 8      8  1  4 0.250000 550 0.000 0.00      1
## 9      9  3  2 0.000000  NA 0.000 0.00      1
## 10     10  5  2 0.000000 580 0.125 0.00      1
## 11     11 10  9 0.203125 625 0.000 0.00      0
## 12     12  5  8 0.281250 375 0.125 0.25      0
## 13     13 10  9 0.203125 400 0.000 0.00      0
## 14     14  5  8 0.281250 355 0.125 0.25      0

parents=length(which(data.ped$parents == 1))

### gametic variance components:
sigma.s.2 = 2552      # gametic variance as father
```

```

sigma.d.2 = 2800      # gametic variances as mother
sigma.sd   = 2670      # gametic covariance
sigma.e.2  = 12756     # residual variance in a gametic model

### construct lambdas for the coefficient matrix (left hand side of mixed model equations):
det.Var = sigma.s.2*sigma.d.2-(sigma.sd*sigma.sd)
la1 = (1/det.Var)*sigma.d.2*sigma.e.2
la2 = (1/det.Var)*(-sigma.sd)*sigma.e.2
la3 = (1/det.Var)*sigma.s.2*sigma.e.2

### construction of the inverse of the gametic relationship matrix with the fortran
### program "GGRInv" (as 3-column lower triangle):
setwd("ProFor") # change to the ProFor directory containing "GGRInv"

# Only parents are included in the gametic relationship matrix. Therefore, a "reduced"
# pedigree that only contains parents must be generated:

ped=data.ped[data.ped$parents == 1, c(which(colnames(data.ped)=="id"),
                                     which(colnames(data.ped)=="dad"),
                                     which(colnames(data.ped)=="mom"),
                                     which(colnames(data.ped)=="F"),
                                     which(colnames(data.ped)=="y"))]

# code the new reduced pedigree:
ped$IdNew = seq(1,nrow(ped))
a = which(ped[,2] == 0)
sirenew = match(ped[,2], ped[,1], nomatch = 0)
pos = which(sirenew == 0)
f = c((dim(ped)[1]+1):((dim(ped)[1])+(length(pos)-length(a))))
sirenew[setdiff(pos,a)] = f
sirenew[a] = 0
ped$IdNewSire = sirenew
a = which(ped[,3] == 0)
damnew = match(ped[,3], ped[,1], nomatch = 0)
pos = which(damnew == 0)
ff = c((f[length(f)]+1):(f[length(f)] + (length(pos)-length(a))))
damnew[setdiff(pos,a)] = ff
damnew[a] = 0
ped$IdNewDam = damnew

# prepare the dataset needed for "GGRInv". The dataset must provide the following
# columns: "id", "dad", "mom", "indicator", "F"
data_in = as.data.frame(cbind(ped$IdNew, ped$IdNewSire, ped$IdNewDam, rep(2, nrow(ped)), ped$F))
colnames(data_in) = c("IdNew", "IdNewSire", "IdNewDam", "indicator", "F")
data_in # note: indicator of 2s -> construction of reduc. gametic relationship matrix G

```

##	IdNew	IdNewSire	IdNewDam	indicator	F
## 1	1	0	0	2	0.000
## 2	2	0	0	2	0.000
## 3	3	1	0	2	0.000
## 4	4	1	0	2	0.000
## 5	5	3	4	2	0.125
## 6	6	1	4	2	0.250
## 7	7	3	2	2	0.000
## 8	8	5	2	2	0.000

```

# write dataset to ProFor directory:
write.table(data_in,"pedigree_input.txt",col.names=FALSE,row.names=FALSE,quote=FALSE,sep="\t")

# run "GGRinv" to construct the inverse of G (AImatrix.giv) with system function:

#####
#
# system("GGRinv > out.txt") #
#
#####

# move inverse of G (AImatrix.giv) to the ASreml directory so it will be available for the
# subsequent analysis with the ASreml software:

# system("mv AImatrix.giv ../ASreml")

### rewrite inverse gametic relationship matrix to a n x n matrix:
setwd("../ASreml") # change to the ASreml directory
GAMinv = read.table("AImatrix.giv", header=FALSE)
GAMi = matrix(ncol = parents*2, nrow = parents*2, 0) # rewrite inv. gam. relationship matrix:
i = 0
for(i in 1:nrow(GAMinv)){
  GAMi[ GAMinv[i,1] , GAMinv[i,2]] = GAMinv[i,3]
  GAMi[ GAMinv[i,2], GAMinv[i,1]] = GAMinv[i,3]
}
GAMi[1:10,1:10]

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]      [,9]      [,10]
## [1,]  2.5  1.5   0   0  -1   0  -1   0  0.0000000  0.0000000
## [2,]  1.5  2.5   0   0  -1   0  -1   0  0.0000000  0.0000000
## [3,]  0.0  0.0   2   1   0   0   0   0  0.0000000  0.0000000
## [4,]  0.0  0.0   1   2   0   0   0   0  0.0000000  0.0000000
## [5,] -1.0 -1.0   0   0   3   1   0   0 -1.0000000  0.0000000
## [6,]  0.0  0.0   0   0   1   2   0   0 -1.0000000  0.0000000
## [7,] -1.0 -1.0   0   0   0   0   3   1  0.0000000 -1.0000000
## [8,]  0.0  0.0   0   0   0   0   1   2  0.0000000 -1.0000000
## [9,]  0.0  0.0   0   0  -1  -1   0   0  2.5714290  0.5714286
## [10,] 0.0  0.0   0   0   0   0  -1  -1  0.5714286  2.5714290

### construction of incidence matrices to set up the mixed model equations:

beob = data.ped[is.na(data.ped$y)==FALSE,1] # IDs of animals with phenotypes

## construct Z.rs:
Z.rs = matrix(ncol = parents*2, nrow = length(beob),0)
Z.rs[1,c(5,6)] = 0.5 # note: records of final progeny are linked to gametes of their sire
Z.rs[2,c(1,2)] = 0.5
Z.rs[5,c(15,16)] = 0.5
Z.rs[6,c(9,10)] = 0.5
Z.rs[7,c(15,16)] = 0.5
Z.rs[8,c(9,10)] = 0.5
Z.rs[3,c(11)] = 1 # note: records of parents are linked to their paternal gamete
Z.rs[4,c(15)] = 1
Z.rs

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,] 0.0 0.0 0 0 0.5 0.5 0 0 0.0 0.0 0 0 0
## [2,] 0.5 0.5 0 0 0.0 0.0 0 0 0.0 0.0 0 0 0
## [3,] 0.0 0.0 0 0 0.0 0.0 0 0 0.0 0.0 1 0 0
## [4,] 0.0 0.0 0 0 0.0 0.0 0 0 0.0 0.0 0 0 0
## [5,] 0.0 0.0 0 0 0.0 0.0 0 0 0.0 0.0 0 0 0
## [6,] 0.0 0.0 0 0 0.0 0.0 0 0 0.5 0.5 0 0 0
## [7,] 0.0 0.0 0 0 0.0 0.0 0 0 0.0 0.0 0 0 0
## [8,] 0.0 0.0 0 0 0.0 0.0 0 0 0.5 0.5 0 0 0
##      [,14] [,15] [,16]
## [1,] 0 0.0 0.0
## [2,] 0 0.0 0.0
## [3,] 0 0.0 0.0
## [4,] 0 1.0 0.0
## [5,] 0 0.5 0.5
## [6,] 0 0.0 0.0
## [7,] 0 0.5 0.5
## [8,] 0 0.0 0.0

## construct Z.rd:
Z.rd = matrix(ncol = parents*2, nrow = length(beob),0)
Z.rd[1,c(7,8)] = 0.5 # note: records of final progeny are linked to gametes of their dam
Z.rd[2,c(7,8)] = 0.5
Z.rd[5,c(13,14)] = 0.5
Z.rd[6,c(11,12)] = 0.5
Z.rd[7,c(13,14)] = 0.5
Z.rd[8,c(11,12)] = 0.5
Z.rd[3,12] = 1 # note: records of parents are linked to their maternal gamete
Z.rd[4,16] = 1

Z.rd

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,] 0 0 0 0 0 0 0 0.5 0.5 0 0 0.0 0.0 0.0
## [2,] 0 0 0 0 0 0 0 0.5 0.5 0 0 0.0 0.0 0.0
## [3,] 0 0 0 0 0 0 0 0.0 0.0 0 0 0.0 1.0 0.0
## [4,] 0 0 0 0 0 0 0 0.0 0.0 0 0 0.0 0.0 0.0
## [5,] 0 0 0 0 0 0 0 0.0 0.0 0 0 0.0 0.0 0.5
## [6,] 0 0 0 0 0 0 0 0.0 0.0 0 0 0.5 0.5 0.0
## [7,] 0 0 0 0 0 0 0 0.0 0.0 0 0 0.0 0.0 0.5
## [8,] 0 0 0 0 0 0 0 0.0 0.0 0 0 0.5 0.5 0.0
##      [,14] [,15] [,16]
## [1,] 0.0 0 0
## [2,] 0.0 0 0
## [3,] 0.0 0 0
## [4,] 0.0 0 1
## [5,] 0.5 0 0
## [6,] 0.0 0 0
## [7,] 0.5 0 0
## [8,] 0.0 0 0

### calculation of weights in W matrix:
ind_beob = as.data.frame(cbind(beob,rep(0, length(beob)))); colnames(ind_beob)=c("ID","index")
m1 = match(ind_beob$ID, data.ped$dad, nomatch = 0)
ind_beob[m1!=0,colnames(ind_beob) == "index"] = 1

```

```

m2 = match(ind_beob$ID, data.ped$mom, nomatch = 0)
ind_beob[m2!=0,colnames(ind_beob) == "index"] = 1
ind_beob = merge(ind_beob, data.ped, by.x="ID", by.y="id")
ind_beob = ind_beob[order(ind_beob$ID),]

W = matrix(ncol=length(beob),nrow=length(beob),0)
for(i in 1:length(beob)){
  if(ind_beob[i,2]==1){
    W[i,i] = 1
  }else{
    W[i,i] = (0.5*(1 - (ind_beob[i,colnames(ind_beob)=="Fs"]))*sigma.s.2 +
              0.5*(1 - (ind_beob[i,colnames(ind_beob)=="Fd"]))*sigma.d.2 + sigma.e.2)/
              (sigma.e.2)
  }
}
W.inv = solve(W)
W.inv

```

```

##           [,1]      [,2] [,3] [,4]      [,5]      [,6]      [,7]      [,8]
## [1,] 0.8265941 0.0000000    0    0 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.8265941    0    0 0.0000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000    1    0 0.0000000 0.0000000 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000    0    1 0.0000000 0.0000000 0.0000000 0.0000000
## [5,] 0.0000000 0.0000000    0    0 0.8265941 0.0000000 0.0000000 0.0000000
## [6,] 0.0000000 0.0000000    0    0 0.0000000 0.8548166 0.0000000 0.0000000
## [7,] 0.0000000 0.0000000    0    0 0.0000000 0.0000000 0.8265941 0.0000000
## [8,] 0.0000000 0.0000000    0    0 0.0000000 0.0000000 0.0000000 0.8548166

```

X: the only fixed effect is the population mean:

```

X = matrix(ncol=1,nrow=length(beob),1)
X

```

```

##           [,1]
## [1,]      1
## [2,]      1
## [3,]      1
## [4,]      1
## [5,]      1
## [6,]      1
## [7,]      1
## [8,]      1

```

y

```

y = as.matrix(data.ped[is.na(data.ped$y)==FALSE,colnames(data.ped)=="y"])
y

```

```

##           [,1]
## [1,] 444
## [2,] 555
## [3,] 550
## [4,] 580
## [5,] 625
## [6,] 375
## [7,] 400
## [8,] 355

```

```

### set up the mixed model equations:

LHS = rbind(cbind((t(X)%*%W.inv%*%X), # left hand side
                  (t(X)%*%W.inv%*%Z.rs),
                  (t(X)%*%W.inv%*%Z.rd)),
            cbind((t(Z.rs)%*%W.inv%*%X),
                  (t(Z.rs)%*%W.inv%*%Z.rs+GAMi*la1),
                  (t(Z.rs)%*%W.inv%*%Z.rd+GAMi*la2)),
            cbind((t(Z.rd)%*%W.inv%*%X),
                  (t(Z.rd)%*%W.inv%*%Z.rs+GAMi*la2),
                  (t(Z.rd)%*%W.inv%*%Z.rd+GAMi*la3)))

RHS = rbind((t(X)%*%W.inv%*%y),t(Z.rs)%*%W.inv%*%y,t(Z.rd)%*%W.inv%*%y) # right hand side

### Solving the mixed model equations by direct inversion of the coefficient matrix
### -> solutions in LS
LS = solve(LHS)%*%RHS

#####
##### ASREML: #####
#####

# Now, the results generated in R (saved in vector LS) should be replicated with ASreml.
# For this purpose, the dataset needed for ASreml can be prepared in R.

# The dataset must contain the columns ID = individual; GeV = paternal gamete of parent
# with phenotype; GeM = maternal gamete of parent with phenotype; GV1 = paternal gamete
# of sire of phenotyped progeny; GV2 = maternal gamete of sire of phenotyped progeny;
# GM1 = paternal gamete of dam of phenotyped progeny; GM2 = maternal gamete of dam of
# phenotyped progeny; y = phenotype; and weight:

ID = ped[,colnames(ped)=="IdNew"]
ped$g1nr = (ID-1)*2 + 1
ped$g2nr = ped$g1nr + 1
## parents with own records:
m=match(ped$id, ind_beob[which(ind_beob$index==1),1], nomatch=0)
dat_sub1 = as.data.frame(cbind(ped[m!=0,colnames(ped)=="id"],
                              ped[m!=0,colnames(ped)=="g1nr"],
                              ped[m!=0,colnames(ped)=="g2nr"],
                              rep(0,nrow(ped[m!=0,])),
                              rep(0,nrow(ped[m!=0,])),
                              rep(0,nrow(ped[m!=0,])),
                              rep(0,nrow(ped[m!=0,])),
                              ped[m!=0,colnames(ped)=="y"]))

## animals with records but no offspring:
ind_beob2=ind_beob[-(which(ind_beob$index==1)),]
ped.sub=ped[,c(1,9,10)]
dad_sub=merge(ind_beob2, ped.sub, by.x="dad", by.y="id")
mom_sub=merge(dad_sub, ped.sub, by.x="mom", by.y="id"); mom_sub=mom_sub[order(mom_sub$ID),]
dat_sub2 = as.data.frame(cbind(mom_sub$ID, rep(0, nrow(mom_sub)),
                              rep(0,nrow(mom_sub)),
                              mom_sub$g1nr.x,

```

```

mom_sub$g2nr.x,
mom_sub$g1nr.y,
mom_sub$g2nr.y,
mom_sub$y))
datasr=rbind(dat_sub1, dat_sub2)
colnames(datasr)=c("ID","GeV","GeM","GV1","GV2","GM1","GM2","y")
datasr=datasr[order(datasr$ID),]
datasr=cbind(datasr, (diag(W.inv)))
colnames(datasr)[9]="weight"
datasr

##   ID GeV GeM GV1 GV2 GM1 GM2   y   weight
## 3   6   0   0   5   6   7   8 444 0.8265941
## 4   7   0   0   1   2   7   8 555 0.8265941
## 1   8  11  12   0   0   0   0 550 1.0000000
## 2  10  15  16   0   0   0   0 580 1.0000000
## 5  11   0   0  15  16  13  14 625 0.8265941
## 6  12   0   0   9  10  11  12 375 0.8548166
## 7  13   0   0  15  16  13  14 400 0.8265941
## 8  14   0   0   9  10  11  12 355 0.8548166

# write dataset (datasr) to ASreml directory:
write.table(datasr, "dataset.txt", col.names=TRUE, row.names=FALSE, quote=FALSE, sep="\t")

### run ASreml from R-script using the system function:

#####
#
# system("asreml -l command_file.as") #
#
#####

#####

### Now, compare results of R and ASreml:
ASR = read.table("command_file.sln", header=FALSE) # read in the ASreml solution file:

result=as.data.frame(cbind(ASR[,3],round(LS,3)))
colnames(result)=c("ASreml_solutions","R_solutions")
result

##   ASreml_solutions R_solutions
## 1          493.400      493.440
## 2           -3.144       -3.144
## 3           -3.144       -3.144
## 4            9.327        9.327
## 5            9.327        9.327
## 6           -6.885       -6.885
## 7           -7.482       -7.482
## 8           -5.586       -5.586
## 9           -4.884       -4.884
## 10          -12.520      -12.520
## 11          -10.570      -10.572
## 12           -5.910       -5.910
## 13           -7.675       -7.675

```

```
## 14          -6.289      -6.289
## 15          10.220      10.222
## 16          -4.460      -4.460
## 17          17.760      17.760
## 18          -3.312      -3.312
## 19          -3.312      -3.312
## 20           9.779       9.779
## 21           9.779       9.779
## 22          -7.225      -7.225
## 23          -7.825      -7.825
## 24          -5.867      -5.867
## 25          -5.109      -5.109
## 26         -13.110     -13.108
## 27         -11.070     -11.071
## 28          -6.230      -6.230
## 29          -8.046      -8.046
## 30          -6.587      -6.587
## 31          10.720      10.718
## 32          -4.677      -4.677
## 33          18.620      18.620
```

```
#####
```

```
# use the solutions to calculate the breeding values and imprinting effects:
```

```
LSi=LS[-1]
```

```
LSasFather=LSi[1:(parents*2)]
```

```
LSasMother=LSi[(parents*2+1):(parents*4)]
```

```
LSasFathergv=LSasFather[seq(1,(parents*2),2)]
```

```
LSasFathergm=LSasFather[seq(2,(parents*2),2)]
```

```
asV=cbind(LSasFathergv,LSasFathergm)
```

```
asV # gametic effects as sire:
```

```
##      LSasFathergv LSasFathergm
## [1,]   -3.144421    -3.144421
## [2,]    9.327352     9.327352
## [3,]   -6.885271    -7.481698
## [4,]   -5.586496    -4.884149
## [5,]  -12.519821   -10.571660
## [6,]   -5.909769    -7.674517
## [7,]   -6.288595     10.22242
## [8,]   -4.460345     17.759815
```

```
LSasMothergv=LSasMother[seq(1,(parents*2),2)]
```

```
LSasMothergm=LSasMother[seq(2,(parents*2),2)]
```

```
asM=cbind(LSasMothergv,LSasMothergm)
```

```
asM # gametic effects as dam:
```

```
##      LSasMothergv LSasMothergm
## [1,]   -3.312229    -3.312229
## [2,]    9.779397     9.779397
## [3,]   -7.224952    -7.825446
## [4,]   -5.866667    -5.108876
## [5,]  -13.108278   -11.070851
```

```
## [6,]    -6.229506    -8.045728
## [7,]    -6.586737    10.717858
## [8,]    -4.676553    18.620332

BVasV=asV[,1]+asV[,2] # breeding value as sire
BVasM=asM[,1]+asM[,2] # breeding value as dam
EIE = BVasM - BVasV    # imprinting effects

cbind(BVasV, BVasM, EIE)

##          BVasV      BVasM      EIE
## [1,]  -6.288843  -6.624458 -0.3356146
## [2,]  18.654704  19.558793  0.9040888
## [3,] -14.366969 -15.050397 -0.6834283
## [4,] -10.470645 -10.975542 -0.5048966
## [5,] -23.091481 -24.179129 -1.0876482
## [6,] -13.584286 -14.275234 -0.6909482
## [7,]   3.933647   4.131120  0.1974731
## [8,]  13.299469  13.943779  0.6443095

setwd("../")
```

Toy example - the generalized gametic model

Inga Blunk

1 Apr 2020

This script provides an example for the *generalized gametic model*. In matrix notation the model is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{hs}\mathbf{a}_s + \mathbf{Z}_{hd}\mathbf{a}_d + \mathbf{e},$$

where $\mathbf{y}$ is a vector of observations, $\boldsymbol{\beta}$ comprises fixed effects and $\mathbf{X}$ is the corresponding incidence matrix. The vectors $\mathbf{a}_s$ and $\mathbf{a}_d$ contain gametic effects under a paternal and maternal expression pattern for phenotyped animals and transmitting abilities for animals without records. Further, the incidence matrices $\mathbf{Z}_{hs}$ and $\mathbf{Z}_{hd}$ link observations to the random gametic effects in $\mathbf{a}_s$ and $\mathbf{a}_d$, while no observation is linked to any of the transmitting abilities in the latter vectors. The covariance of random effects is assumed to be:

$$\text{Var} \begin{bmatrix} \mathbf{a}_s \\ \mathbf{a}_d \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{G}}\sigma_s^2 & \bar{\mathbf{G}}\sigma_{sd} & 0 \\ \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_d^2 & 0 \\ 0 & 0 & \mathbf{I}\sigma_e^2 \end{bmatrix}.$$

The *generalized gametic model* uses a generalized gametic relationship matrix $\bar{\mathbf{G}}$ instead of the classical gametic relationship matrix $\mathbf{G}$ (Schaeffer *et al.*, 1989) so that gametic effects are predicted for phenotyped individuals and transmitting abilities for animals without own observations. The gametic variances are used in the mixed model equations as:

$$\begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_2 & \lambda_3 \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & \sigma_{sd} \\ \sigma_{sd} & \sigma_d^2 \end{bmatrix}^{-1} \sigma_e^2.$$

The mixed model equations are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z}_{hs} & \mathbf{X}'\mathbf{Z}_{hd} \\ \mathbf{Z}_{hs}'\mathbf{X} & \mathbf{Z}_{hs}'\mathbf{Z}_{hs} + \bar{\mathbf{G}}^{-1}\lambda_1 & \mathbf{Z}_{hs}'\mathbf{Z}_{hd} + \bar{\mathbf{G}}^{-1}\lambda_2 \\ \mathbf{Z}_{hd}'\mathbf{X} & \mathbf{Z}_{hd}'\mathbf{Z}_{hs} + \bar{\mathbf{G}}^{-1}\lambda_2 & \mathbf{Z}_{hd}'\mathbf{Z}_{hd} + \bar{\mathbf{G}}^{-1}\lambda_3 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{a}}_s \\ \hat{\mathbf{a}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}_{hs}'\mathbf{y} \\ \mathbf{Z}_{hd}'\mathbf{y} \end{bmatrix}.$$

Solving the mixed model equations by direct inversion of the coefficient matrix, provides $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{a}}_s$ and $\hat{\mathbf{a}}_d$.

References

- Gilmour, A.R., Gogel, B.J., Cullis, B.R., and Thompson R. 2009. ASReml user guide release 3.0. VSN International Ltd., Hemel Hempstead, UK. <https://asreml.kb.vsnr.co.uk/wp-content/uploads/sites/3/2018/02/ASReml-3-User-Guide.pdf>
- Schaeffer, L.R., Kennedy, B.W., and Gibson, J.P. 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Example

In the following a toy example is provided in an R environment, i.e. based on a toy dataset the mixed model equations of the *generalized gametic model* will be solved in R. Subsequently, the same dataset will be analyzed with ASReml to show how the model can be implemented into the software. Eventually, the R and ASReml solutions will be compared. At the end of the R script breeding values and imprinting effects will be calculated. They can be compared to the results generated with the other corresponding *gametic models* in order to show their equivalences. The fortran program *GGRinv* that builds the inverse generalized gametic relationship matrix $\bar{\mathbf{G}}$ and the ASReml software (version 3.0; Gilmour *et al.*, 2009) can be navigated from the following R script. The corresponding command files (command_file.txt for *GGRinv*; command_file.as for

ASreml) must be available within the corresponding directories, which are *ProFor* and *ASreml*, respectively. The example was build in an UNIX environment.

```
### Overall number of animals in the pedigree:

nr_animals = 14

### data file:
# F=inbreeding coefficient; y=phenotype
data.ped = data.frame(id = seq(1,nr_animals,1),
                      dad = c(0,0,1,1,3,3,1,1,3,5,10,5,10,5),
                      mom = c(0,0,0,0,4,4,4,4,2,2,9,8,9,8),
                      F = c(0,0,0,0,0.125,0.125,0.250,0.250,0,0,0.203125,0.28125,0.203125,0.28125),
                      y = c(NA,NA,NA,NA,NA,444,555,550,NA,580,625,375,400,355))

beob = data.ped[is.na(data.ped$y)==FALSE,1]
data.ped # toy data to be analyzed in the following:

##      id dad mom      F    y
## 1    1  0  0 0.000000  NA
## 2    2  0  0 0.000000  NA
## 3    3  1  0 0.000000  NA
## 4    4  1  0 0.000000  NA
## 5    5  3  4 0.125000  NA
## 6    6  3  4 0.125000 444
## 7    7  1  4 0.250000 555
## 8    8  1  4 0.250000 550
## 9    9  3  2 0.000000  NA
## 10   10  5  2 0.000000 580
## 11   11 10  9 0.203125 625
## 12   12  5  8 0.281250 375
## 13   13 10  9 0.203125 400
## 14   14  5  8 0.281250 355

### gametic variance components:
sigma.s.2 = 2552      # gametic variance as father
sigma.d.2 = 2800      # gametic variances as mother
sigma.sd = 2670       # gametic covariance
sigma.e.2 = 12756     # residual variance in a gametic model

### construct lambdas for the coefficient matrix (left hand side of mixed model equations):
det.Var = sigma.s.2*sigma.d.2-(sigma.sd*sigma.sd)
la1 = (1/det.Var)*sigma.d.2*sigma.e.2
la2 = (1/det.Var)*(-sigma.sd)*sigma.e.2
la3 = (1/det.Var)*sigma.s.2*sigma.e.2

### construction of the inverse of the gametic relationship matrix with the fortran
### program "GGRInv" (as 3-column lower triangle):
setwd("ProFor") # change to the ProFor directory containing "GGRInv"

# prepare the dataset needed for "GGRInv". The dataset must provide the following
# columns: "id","dad","mom","indicator","F"
data_in = cbind(data.ped[,1:3],rep(1,nrow(data.ped)),data.ped$F)
colnames(data_in)[c(4,5)] = c("indicator","F")
```

```

data_in[is.na(data.ped$y) == FALSE, colnames(data_in) == "indicator"] = 2
data_in # note the indicator vector: 2s for gametic effects and 1s for transmitting ability

##      id dad mom indicator      F
## 1    1   0   0          1 0.000000
## 2    2   0   0          1 0.000000
## 3    3   1   0          1 0.000000
## 4    4   1   0          1 0.000000
## 5    5   3   4          1 0.125000
## 6    6   3   4          2 0.125000
## 7    7   1   4          2 0.250000
## 8    8   1   4          2 0.250000
## 9    9   3   2          1 0.000000
## 10  10   5   2          2 0.000000
## 11  11  10   9          2 0.203125
## 12  12   5   8          2 0.281250
## 13  13  10   9          2 0.203125
## 14  14   5   8          2 0.281250

# write dataset to ProFor directory:
write.table(data_in, "pedigree_input.txt", col.names=FALSE, row.names=FALSE, quote=FALSE, sep="\t")

# run "GGRinv" to construct the inverse of G (AImatrix.giv) with system function:

#####
#
# system("GGRinv > out.txt") #
#
#####

# move inverse of G (AImatrix.giv) to the ASreml directory so it will be available for the
# subsequent analysis with the ASreml software:

# system("mv AImatrix.giv ../ASreml")

### read in cross reference table (output file of fortran program):
cross = read.table("crossref.txt", header = FALSE)
anzgl = max(max(cross$V4), max(cross$V5))
### rewrite inverse generalized gametic relationship matrix to a n x n matrix:
setwd("../ASreml") # change to the ASreml directory
GAMinv = read.table("AImatrix.giv", header=FALSE) # 3-col low triangle of inv. G_bar
GAMi = matrix(ncol=anzgl, nrow=anzgl, 0) # inverse of G_bar as n x n matrix
i = 0
for(i in 1:nrow(GAMinv)){
  GAMi[ GAMinv[i,1] , GAMinv[i,2]] = GAMinv[i,3]
  GAMi[ GAMinv[i,2], GAMinv[i,1]] = GAMinv[i,3]
}
GAMi[1:10,1:10]

##      [,1] [,2]      [,3]      [,4]      [,5] [,6] [,7] [,8] [,9]
## [1,] 7.333333 0 -1.333333 -1.333333 0.00000 0 0 -2 0
## [2,] 0.000000 5 1.000000 0.000000 0.00000 0 0 0 0
## [3,] -1.333333 1 6.666667 1.000000 -2.00000 -2 0 0 0
## [4,] -1.333333 0 1.000000 9.666667 -2.00000 0 -2 0 -2

```

```

## [5,] 0.000000 0 -2.000000 -2.000000 10.85714 0 0 0 0
## [6,] 0.000000 0 -2.000000 0.000000 0.00000 2 0 0 0
## [7,] 0.000000 0 0.000000 -2.000000 0.00000 0 2 0 0
## [8,] -2.000000 0 0.000000 0.000000 0.00000 0 0 2 0
## [9,] 0.000000 0 0.000000 -2.000000 0.00000 0 0 0 2
## [10,] -2.000000 0 0.000000 0.000000 0.00000 0 0 0 0
## [,10]
## [1,] -2.000000
## [2,] 0.000000
## [3,] 0.000000
## [4,] 0.000000
## [5,] 0.000000
## [6,] 0.000000
## [7,] 0.000000
## [8,] 0.000000
## [9,] 0.000000
## [10,] 3.333333

### construction of incidence matrices to set up the mixed model equations:
data.ped = cbind(data.ped, cross$V4, cross$V5)
colnames(data.ped) = c("id","dad","mom","F","y","z1","z2")
Z.hs = matrix(ncol = ncol(GAMi), nrow = length(which(is.na(data.ped$y) == FALSE)), 0)
Z.hd = matrix(ncol = ncol(GAMi), nrow = length(which(is.na(data.ped$y) == FALSE)), 0)
xc = 1; xr = 1
for(i in 1:nrow(data.ped)){
  if(data.ped[i,colnames(data.ped) == "z1"] ==
    data.ped[i,colnames(data.ped) == "z2"]){ xc = xc + 1
  if(data.ped[i,colnames(data.ped) == "z1"] !=
    data.ped[i,colnames(data.ped) == "z2"]){
    Z.hs[xr,xc] = 1; xc = xc + 1
    Z.hd[xr,xc] = 1; xc = xc + 1
    xr = xr + 1
  }
}
}
Z.hs # note that records of animals are linked to their expressed paternal gametes

```

```

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,] 0 0 0 0 0 1 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 1 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 1 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 1
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## [1,] 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0
## [5,] 0 1 0 0 0 0 0 0 0
## [6,] 0 0 0 1 0 0 0 0 0
## [7,] 0 0 0 0 0 1 0 0 0
## [8,] 0 0 0 0 0 0 0 1 0

```

```
Z.hd # note that records of animals are linked to their expressed maternal gametes:
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]  0    0    0    0    0    0    1    0    0    0    0    0    0
## [2,]  0    0    0    0    0    0    0    0    1    0    0    0    0
## [3,]  0    0    0    0    0    0    0    0    0    0    1    0    0
## [4,]  0    0    0    0    0    0    0    0    0    0    0    0    0
## [5,]  0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]  0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]  0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]  0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## [1,]  0     0     0     0     0     0     0     0     0
## [2,]  0     0     0     0     0     0     0     0     0
## [3,]  0     0     0     0     0     0     0     0     0
## [4,]  1     0     0     0     0     0     0     0     0
## [5,]  0     0     1     0     0     0     0     0     0
## [6,]  0     0     0     0     1     0     0     0     0
## [7,]  0     0     0     0     0     0     1     0     0
## [8,]  0     0     0     0     0     0     0     0     1
```

```
## X: the only fixed effect is the population mean:
```

```
X = matrix(ncol = 1, nrow = length(beob),1);X
```

```
##      [,1]
## [1,]  1
## [2,]  1
## [3,]  1
## [4,]  1
## [5,]  1
## [6,]  1
## [7,]  1
## [8,]  1
```

```
## y
```

```
y = as.matrix(data.ped[which(is.na(data.ped$y) == FALSE), which(colnames(data.ped) == "y")]);y
```

```
##      [,1]
## [1,] 444
## [2,] 555
## [3,] 550
## [4,] 580
## [5,] 625
## [6,] 375
## [7,] 400
## [8,] 355
```

```
### set up the mixed model equations:
```

```
LHS = rbind(cbind((t(X)%*%X), # left hand side (coefficient matrix)
                  (t(X)%*%Z.hs),
                  (t(X)%*%Z.hd)),
            cbind((t(Z.hs)%*%X),
                  (t(Z.hs)%*%Z.hs+GAMi*la1),
                  (t(Z.hs)%*%Z.hd+GAMi*la2)),
            cbind((t(Z.hd)%*%X),
                  (t(Z.hd)%*%Z.hs+GAMi*la2),
```

```

(t(Z.hd)%*%Z.hd+GAMi*1a3)))

RHS = rbind((t(X)%*%y),t(Z.hs)%*%y,t(Z.hd)%*%y) # right hand side

### Solving the mixed model equations by direct inversion of the coefficient matrix
### -> solutions in LS
LS = solve(LHS)%*%RHS

#####
##### ASREML: #####
#####

# Now, the results generated in R (saved in vector LS) should be replicated with ASreml.
# For this purpose, the dataset needed for ASreml can be prepared in R.

# The dataset must contain the columns ID = individual; g1nr = paternal gamete; g2nr =
# maternal gamete; and the phenotype:
data.ped[(data.ped$z1/data.ped$z2) == Inf, colnames(data.ped) == "z2"] =
  data.ped[(data.ped$z1/data.ped$z2) == Inf, colnames(data.ped) == "z1"]
PP = as.data.frame(cbind(data.ped$id, data.ped$z1, data.ped$z2, data.ped$y))
colnames(PP) = c("ID", "g1nr", "g2nr", "phenotype")
PP[is.na(PP$phenotype) == TRUE, colnames(PP) == "phenotype"] = 0
PP

##      ID g1nr g2nr phenotype
## 1      1      1      1          0
## 2      2      2      2          0
## 3      3      3      3          0
## 4      4      4      4          0
## 5      5      5      5          0
## 6      6      6      7      444
## 7      7      8      9      555
## 8      8     10     11      550
## 9      9     12     12          0
## 10     10     13     14      580
## 11     11     15     16      625
## 12     12     17     18      375
## 13     13     19     20      400
## 14     14     21     22      355

# note that for animals without phenotypes the same entry (gamete number) is used at the
# position of the paternal and maternal gametes!
# write dataset (PP) to ASreml directory:
write.table(PP,"dataset.txt",col.names = TRUE,row.names=FALSE,quote = FALSE,sep = "\t")

### run ASreml from R-script using the system function:

#####
#
# system("asreml -l command_file.as") #
#
#####

#####

```

```

### Now, compare results of R and ASreml:
ASR = read.table("command_file.sln", header=FALSE) # read in the ASreml solution file:

result=as.data.frame(cbind(round(ASR[,3],3),round(LS,3)))
colnames(result)=c("ASreml_solution","R_solution")
result

```

##	ASreml_solution	R_solution
## 1	493.400	493.440
## 2	-3.144	-3.144
## 3	9.327	9.327
## 4	-7.183	-7.183
## 5	-5.235	-5.235
## 6	-11.550	-11.546
## 7	-10.220	-10.224
## 8	-8.416	-8.416
## 9	2.659	2.659
## 10	0.837	0.837
## 11	-5.910	-5.910
## 12	-7.675	-7.675
## 13	1.967	1.967
## 14	-4.460	-4.460
## 15	17.760	17.760
## 16	16.810	16.807
## 17	12.590	12.594
## 18	-19.010	-19.010
## 19	-13.490	-13.485
## 20	-1.797	-1.797
## 21	-6.870	-6.870
## 22	-20.510	-20.506
## 23	-14.830	-14.827
## 24	-3.312	-3.312
## 25	9.779	9.779
## 26	-7.525	-7.525
## 27	-5.488	-5.488
## 28	-12.090	-12.090
## 29	-10.710	-10.706
## 30	-8.823	-8.823
## 31	2.760	2.760
## 32	0.880	0.880
## 33	-6.229	-6.229
## 34	-8.046	-8.046
## 35	2.066	2.066
## 36	-4.677	-4.677
## 37	18.620	18.620
## 38	17.600	17.599
## 39	13.210	13.210
## 40	-19.900	-19.898
## 41	-14.160	-14.157
## 42	-1.865	-1.865
## 43	-7.202	-7.202
## 44	-21.460	-21.464
## 45	-15.560	-15.564

```
#####

# use the solutions to calculate the breeding values and imprinting effects:
LS0 = LS[-1]
LSasV = LS0[1:anzgl]
LSasM = LS0[(anzgl+1):(anzgl*2)]
genEff = as.data.frame(cbind(LSasV, LSasM))
genEff$eff.art = rep(NA,nrow(genEff)); Za = 1
for(i in 1:nrow(data.ped)){
  if(is.na(data.ped[i, colnames(data.ped) == "y"])) == TRUE){
    genEff[Za, colnames(genEff) == "eff.art"] = "TA"
    Za = Za + 1
  }
  if(is.na(data.ped[i, colnames(data.ped) == "y"])) == FALSE){
    genEff[Za, colnames(genEff) == "eff.art"] = "Gam"
    genEff[Za + 1, colnames(genEff) == "eff.art"] = "Gam"
    Za = Za + 2
  }
}
breedingsvalues = as.data.frame(matrix(nrow = nrow(data.ped), ncol = 2, NA))
colnames(breedingsvalues) = c("BVasV","BVasM")
Za = 1
for(i in 1:nrow(data.ped)){
  if(genEff[Za, colnames(genEff) == "eff.art"] == "TA"){
    breedingsvalues[i,1] = genEff[Za, colnames(genEff) == "LSasV"]*2
    breedingsvalues[i,2] = genEff[Za, colnames(genEff) == "LSasM"]*2
    Za = Za + 1
  } else{
    breedingsvalues[i,1] = genEff[Za, colnames(genEff) == "LSasV"] +
      genEff[Za+1, colnames(genEff) == "LSasV"]
    breedingsvalues[i,2] = genEff[Za, colnames(genEff) == "LSasM"] +
      genEff[Za+1, colnames(genEff) == "LSasM"]
    Za = Za + 2
  }
}
EIE = breedingsvalues$BVasM - breedingsvalues$BVasV # imprinting effects
cbind(breedingsvalues,EIE)
```

##	BVasV	BVasM	EIE
## 1	-6.288837	-6.624451	-0.3356143
## 2	18.654707	19.558796	0.9040889
## 3	-14.366976	-15.050404	-0.6834286
## 4	-10.470645	-10.975541	-0.5048965
## 5	-23.091502	-24.179151	-1.0876491
## 6	-18.639887	-19.529497	-0.8896101
## 7	3.496317	3.640072	0.1437550
## 8	-13.584271	-14.275219	-0.6909475
## 9	3.933646	4.131119	0.1974730
## 10	13.299462	13.943772	0.6443092
## 11	29.401096	30.809074	1.4079778
## 12	-32.495030	-34.055335	-1.5603044
## 13	-8.667527	-9.067481	-0.3999538
## 14	-35.333362	-37.028195	-1.6948328

```
setwd("../")
```

Toy example - the reduced generalized gametic model

Inga Blunk

1 Apr 2020

This script provides an example for the *reduced generalized gametic model*. In matrix notation the model is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{rhs}\mathbf{a}_s + \mathbf{Z}_{rhd}\mathbf{a}_d + \boldsymbol{\epsilon},$$

where $\mathbf{y}$ is a vector of observations, $\boldsymbol{\beta}$ comprises fixed effects and $\mathbf{X}$ is the corresponding incidence matrix. The genetic effect vectors $\mathbf{a}_s$ and $\mathbf{a}_d$ contain gametic effects for phenotyped parents and transmitting abilities for parents without own observations. Final progeny do not obtain a genetic effect. The vector $\boldsymbol{\epsilon}$ is a vector of residuals, which either is $\mathbf{r}$ for records from final progeny that are linked to the genetic effects of their parents or $\mathbf{e}$ for records from parents represented by their own two gametic effects. Further, the incidence matrices $\mathbf{Z}_{rhs}$ and $\mathbf{Z}_{rhd}$ link observations with the genetic effects (transmitting abilities and gametic effects). If observations originate from individuals without offspring, i.e. from final offspring, the incidence matrices link the observations to the transmitting abilities of their parents (entry of 1), if these parents do not have records on their own. If the parents are phenotyped, entries of 0.5 at the positions of both parental gametic effects are needed. If observations originate from individuals with offspring an entry of 1 is needed in $\mathbf{Z}_{rhs}$ and $\mathbf{Z}_{rhd}$ at the position of their gametes, respectively. The covariance of random effects is assumed to be:

$$Var \begin{bmatrix} \mathbf{a}_s \\ \mathbf{a}_d \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{G}}\sigma_s^2 & \bar{\mathbf{G}}\sigma_{sd} & 0 \\ \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_d^2 & 0 \\ 0 & 0 & \mathbf{W}\sigma_e^2 \end{bmatrix}.$$

The model uses the generalized gametic relationship matrix $\bar{\mathbf{G}}$ instead of the classical gametic relationship matrix $\mathbf{G}$ (Schaeffer *et al.*, 1989). The matrix $\bar{\mathbf{G}}$ only contains the relationships between parents and their ancestors. Matrix $\mathbf{W}$ is a diagonal matrix containing weightings of observations. For observations from parents the diagonal elements are $w = 1$. For observations from final progeny w is:

$$\frac{0.5\sigma_s^2(1 - F_{si}) + 0.5\sigma_d^2(1 - F_{di}) + \sigma_e^2}{\sigma_e^2},$$

where F_{si} and F_{di} are the inbreeding coefficients of parents of the phenotyped final progeny. The gametic variances are used in the mixed model equations as:

$$\begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_2 & \lambda_3 \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & \sigma_{sd} \\ \sigma_{sd} & \sigma_d^2 \end{bmatrix}^{-1} \sigma_e^2.$$

The mixed model equations are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{W}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{W}^{-1}\mathbf{Z}_{rhs} & \mathbf{X}'\mathbf{W}^{-1}\mathbf{Z}_{rhd} \\ \mathbf{Z}_{rhs}'\mathbf{W}^{-1}\mathbf{X} & \mathbf{Z}_{rhs}'\mathbf{W}^{-1}\mathbf{Z}_{rhs} + \bar{\mathbf{G}}_r^{-1}\lambda_1 & \mathbf{Z}_{rhs}'\mathbf{W}^{-1}\mathbf{Z}_{rhd} + \bar{\mathbf{G}}_r^{-1}\lambda_2 \\ \mathbf{Z}_{rhd}'\mathbf{W}^{-1}\mathbf{X} & \mathbf{Z}_{rhd}'\mathbf{W}^{-1}\mathbf{Z}_{rhs} + \bar{\mathbf{G}}_r^{-1}\lambda_2 & \mathbf{Z}_{rhd}'\mathbf{W}^{-1}\mathbf{Z}_{rhd} + \bar{\mathbf{G}}_r^{-1}\lambda_3 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{a}}_s \\ \hat{\mathbf{a}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{W}^{-1}\mathbf{y} \\ \mathbf{Z}_{rhs}'\mathbf{W}^{-1}\mathbf{y} \\ \mathbf{Z}_{rhd}'\mathbf{W}^{-1}\mathbf{y} \end{bmatrix}.$$

Solving the mixed model equations by direct inversion of the coefficient matrix, provides $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{a}}_s$ and $\hat{\mathbf{a}}_d$.

References

- Gilmour, A.R., Gogel, B.J., Cullis, B.R., and Thompson R. 2009. ASReml user guide release 3.0. VSN International Ltd., Hemel Hempstead, UK. <https://asreml.kb.vsnl.co.uk/wp-content/uploads/sites/3/2018/02/ASReml-3-User-Guide.pdf>
- Schaeffer, L.R., Kennedy, B.W., and Gibson, J.P. 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Example

In the following a toy example is provided in an R environment, i.e. based on a toy dataset the mixed model equations of the *reduced generalized gametic model* will be solved in R. Subsequently, the same dataset will be analyzed with ASreml to show how the model can be implemented into the software. Eventually, the R and ASreml solutions will be compared. At the end of the R script breeding values and imprinting effects will be calculated. They can be compared to the results generated with the other corresponding *gametic models* in order to show their equivalences. The fortran program *GGRinv* that builds the inverse generalized gametic relationship matrix $\bar{G}$ and the ASreml software (version 3.0; Gilmour *et al.*, 2009) can be navigated from the following R script. The corresponding command files (command_file.txt for *GGRinv*; command_file.as for ASreml) must be available within the corresponding directories, which are *ProFor* and *ASreml*, respectively. The example was build in an UNIX environment.

```
### Overall number of animals in the pedigree:

nr_animals = 14

### data file:
# F = inbreeding coefficient; Fs = F of father; Fd = F of mother; y = phenotype
data.ped = data.frame(
  id = seq(1,nr_animals,1),
  dad = c(0,0,1,1,3,3,1,1,3,5,10,5,10,5),
  mom = c(0,0,0,0,4,4,4,4,2,2,9,8,9,8),
  F = c(0,0,0,0,0.125,0.125,0.250,0.250,0,0,0.203125,0.28125,0.203125,0.28125),
  y = c(NA,NA,NA,NA,NA,444,555,550,NA,580,625,375,400,355),
  Fs = c(0,0,0,0,0,0,0,0,0,0.125,0,0.125,0,0.125),
  Fd = c(0,0,0,0,0,0,0,0,0,0,0.25,0,0.25))

# who are the parents?
data.ped$eltern = rep(0,nrow(data.ped))
# sires:
m = match(data.ped$id, data.ped$dad, nomatch=0)
data.ped[m!=0,colnames(data.ped) == "eltern"] = 1
# dams:
m = match(data.ped$id, data.ped$mom, nomatch=0)
data.ped[m!=0,colnames(data.ped) == "eltern"] = 1

data.ped # toy data to be analyzed in the following:
```

##	id	dad	mom	F	y	Fs	Fd	eltern
## 1	1	0	0	0.000000	NA	0.000	0.00	1
## 2	2	0	0	0.000000	NA	0.000	0.00	1
## 3	3	1	0	0.000000	NA	0.000	0.00	1
## 4	4	1	0	0.000000	NA	0.000	0.00	1
## 5	5	3	4	0.125000	NA	0.000	0.00	1
## 6	6	3	4	0.125000	444	0.000	0.00	0
## 7	7	1	4	0.250000	555	0.000	0.00	0
## 8	8	1	4	0.250000	550	0.000	0.00	1
## 9	9	3	2	0.000000	NA	0.000	0.00	1
## 10	10	5	2	0.000000	580	0.125	0.00	1
## 11	11	10	9	0.203125	625	0.000	0.00	0
## 12	12	5	8	0.281250	375	0.125	0.25	0
## 13	13	10	9	0.203125	400	0.000	0.00	0
## 14	14	5	8	0.281250	355	0.125	0.25	0

```

### gametic variance components:
sigma.s.2 = 2552      # gametic variance as father
sigma.d.2 = 2800      # gametic variances as mother
sigma.sd   = 2670      # gametic covariance
sigma.e.2  = 12756     # residual variance in a gametic model

### construct lambdas for the coefficient matrix (left hand side of mixed model equations):
det.Var = sigma.s.2*sigma.d.2-(sigma.sd*sigma.sd)
la1 = (1/det.Var)*sigma.d.2*sigma.e.2
la2 = (1/det.Var)*(-sigma.sd)*sigma.e.2
la3 = (1/det.Var)*sigma.s.2*sigma.e.2

### construction of the inverse of the gametic relationship matrix with the fortran
### program "GGRinv" (as 3-column lower triangle):
setwd("ProFor") # change to the ProFor directory containing "GGRinv"

# Only parents are included in the gametic relationship matrix. Therefore, a "reduced"
# pedigree that only contains parents must be generated:
ped = data.ped[data.ped$eltern == 1, c(which(colnames(data.ped) == "id"),
                                     which(colnames(data.ped) == "dad"),
                                     which(colnames(data.ped) == "mom"),
                                     which(colnames(data.ped) == "F"),
                                     which(colnames(data.ped) == "y"))]

# code the new reduced pedigree:
ped$IdNew = seq(1,nrow(ped))
options(scipen = 999)
a = which(ped[,2] == 0)
sirenew = match(ped[,2],ped[,1],nomatch = 0)
pos = which(sirenew == 0)
f = c((dim(ped)[1]+1):((dim(ped)[1])+(length(pos)-length(a))))
sirenew[setdiff(pos,a)] = f
sirenew[a] = 0
ped$IdNewSire = sirenew
a = which(ped[,3] == 0)
damnew = match(ped[,3],ped[,1],nomatch = 0)
pos = which(damnew == 0)
ff = c((f[length(f)]+1):(f[length(f)] + (length(pos)-length(a))))
damnew[setdiff(pos,a)] = ff
damnew[a] = 0
ped$IdNewDam = damnew

# prepare the dataset needed for "GGRinv". The dataset must provide the following
# columns: "id", "dad", "mom", "indicator", "F"
data_in = as.data.frame(cbind(ped$IdNew,ped$IdNewSire,ped$IdNewDam,rep(1,nrow(ped)),ped$F))
colnames(data_in) = c("IdNew","IdNewSire","IdNewDam","indicator","F")
data_in[is.na(ped$y) == FALSE, colnames(data_in) == "indicator"] = 2
data_in # note the indicator vector: 2s for gametic effects and 1s for transmitting ability

```

##	IdNew	IdNewSire	IdNewDam	indicator	F
## 1	1	0	0	1	0.000
## 2	2	0	0	1	0.000
## 3	3	1	0	1	0.000
## 4	4	1	0	1	0.000
## 5	5	3	4	1	0.125

```
## 6      6      1      4      2 0.250
## 7      7      3      2      1 0.000
## 8      8      5      2      2 0.000
```

```
# write dataset to ProFor directory:
write.table(data_in, "pedigree_input.txt", col.names=FALSE, row.names=FALSE, quote=FALSE, sep="\t")

# run "GGRinv" to construct the inverse of G (AImatrix.giv) with system function:

#####
#
# system("GGRinv > out.txt") #
#
#####

# move inverse of G (AImatrix.giv) to the ASReml directory so it will be available for the
# subsequent analysis with the ASReml software:

# system("mv AImatrix.giv ../ASReml")

### rewrite inverse gametic relationship matrix to a n x n matrix:
setwd("../ASReml") # change to the ASReml directory

GAMinv = read.table("AImatrix.giv", header = FALSE)
GAMi = matrix(ncol = sum(data_in$indicator), nrow = sum(data_in$indicator), 0)
i = 0
for(i in 1:nrow(GAMinv)){
  GAMi[GAMinv[i,1],GAMinv[i,2]] = GAMinv[i,3]
  GAMi[GAMinv[i,2],GAMinv[i,1]] = GAMinv[i,3]
}
GAMi[1:10,1:10]
```

```
##      [,1] [,2]      [,3]      [,4]      [,5] [,6] [,7] [,8]
## [1,] 5.333333 0 -1.333333 -1.333333 0.000000 -2 0 0
## [2,] 0.000000 5 1.000000 0.000000 0.000000 0 0 -2
## [3,] -1.333333 1 4.666667 1.000000 -2.000000 0 0 -2
## [4,] -1.333333 0 1.000000 5.666667 -2.000000 0 -2 0
## [5,] 0.000000 0 -2.000000 -2.000000 6.285714 0 0 0
## [6,] -2.000000 0 0.000000 0.000000 0.000000 2 0 0
## [7,] 0.000000 0 0.000000 -2.000000 0.000000 0 2 0
## [8,] 0.000000 -2 -2.000000 0.000000 0.000000 0 0 4
## [9,] 0.000000 0 0.000000 0.000000 -2.285714 0 0 0
## [10,] 0.000000 -2 0.000000 0.000000 0.000000 0 0 0
##      [,9] [,10]
## [1,] 0.000000 0
## [2,] 0.000000 -2
## [3,] 0.000000 0
## [4,] 0.000000 0
## [5,] -2.285714 0
## [6,] 0.000000 0
## [7,] 0.000000 0
## [8,] 0.000000 0
## [9,] 2.285714 0
## [10,] 0.000000 2
```

```
### construction of incidence matrices to set up the mixed model equations:
```

```
beob = data.ped[which(is.na(data.ped$y) == FALSE),1]
Z.rhs = matrix(ncol = sum(data_in$indicator),nrow = length(beob),0)
Z.rhd = matrix(ncol = sum(data_in$indicator),nrow = length(beob),0)
```

```
## Z.rhs
```

```
Z.rhs[1,3] = 1
Z.rhs[2,1] = 1
Z.rhs[3,6] = 1
Z.rhs[4,9] = 1
Z.rhs[5,c(9,10)] = 0.5
Z.rhs[6,5] = 1
Z.rhs[7,c(9,10)] = 0.5
Z.rhs[8,5] = 1
```

```
# note: If observations originate from final offspring the incidence matrices link the
# observations to the transmitting abilities of their parents (entry of 1), if these
# parents do not have records. If the parents are phenotyped, entries of 0.5 at the
# positions of both parental gametic effects are needed. If observations originate from
# individuals with offspring an entry of 1 is needed
```

```
Z.rhs
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    0    0    1    0    0    0    0    0    0 0.0    0.0
## [2,]    1    0    0    0    0    0    0    0    0 0.0    0.0
## [3,]    0    0    0    0    0    0    1    0    0 0.0    0.0
## [4,]    0    0    0    0    0    0    0    0    0 1.0    0.0
## [5,]    0    0    0    0    0    0    0    0    0 0.5    0.5
## [6,]    0    0    0    0    1    0    0    0    0 0.0    0.0
## [7,]    0    0    0    0    0    0    0    0    0 0.5    0.5
## [8,]    0    0    0    0    1    0    0    0    0 0.0    0.0
```

```
## Z.rhd
```

```
Z.rhd[1,4] = 1
Z.rhd[2,4] = 1
Z.rhd[3,7] = 1
Z.rhd[4,10] = 1
Z.rhd[5,8] = 1
Z.rhd[6,c(6,7)] = 0.5
Z.rhd[7,8] = 1
Z.rhd[8,c(6,7)] = 0.5
Z.rhd
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    0    0    0    1    0 0.0 0.0    0    0    0
## [2,]    0    0    0    1    0 0.0 0.0    0    0    0
## [3,]    0    0    0    0    0 0.0 1.0    0    0    0
## [4,]    0    0    0    0    0 0.0 0.0    0    0    1
## [5,]    0    0    0    0    0 0.0 0.0    1    0    0
## [6,]    0    0    0    0    0 0.5 0.5    0    0    0
## [7,]    0    0    0    0    0 0.0 0.0    1    0    0
## [8,]    0    0    0    0    0 0.5 0.5    0    0    0
```

```
### calculation of weights in W matrix:
```

```
ind_beob = as.data.frame(cbind(beob,rep(0, length(beob))))
```

```

colnames(ind_beob) = c("ID", "index")
m1 = match(ind_beob$ID, data.ped$dad, nomatch = 0)
ind_beob[m1!=0, colnames(ind_beob) == "index"] = 1
m2 = match(ind_beob$ID, data.ped$mom, nomatch = 0)
ind_beob[m2!=0, colnames(ind_beob) == "index"] = 1
ind_beob = merge(ind_beob, data.ped, by.x = "ID", by.y = "id")
ind_beob = ind_beob[order(ind_beob$ID),]
W = matrix(ncol = length(beob), nrow = length(beob), 0)
for(i in 1:length(beob)){
  if(ind_beob[i,2] == 1){
    W[i,i] = 1
  }else{
    W[i,i] = (0.5*(1 - (ind_beob[i, colnames(ind_beob) == "Fs"]))*sigma.s.2 +
              0.5*(1 - (ind_beob[i, colnames(ind_beob) == "Fd"]))*sigma.d.2 + sigma.e.2)/
              (sigma.e.2)
  }
}
W.inv = solve(W)
W.inv

```

```

##           [,1]      [,2] [,3] [,4]      [,5]      [,6]      [,7]      [,8]
## [1,] 0.8265941 0.0000000    0    0 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.8265941    0    0 0.0000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000    1    0 0.0000000 0.0000000 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000    0    1 0.0000000 0.0000000 0.0000000 0.0000000
## [5,] 0.0000000 0.0000000    0    0 0.8265941 0.0000000 0.0000000 0.0000000
## [6,] 0.0000000 0.0000000    0    0 0.0000000 0.8548166 0.0000000 0.0000000
## [7,] 0.0000000 0.0000000    0    0 0.0000000 0.0000000 0.8265941 0.0000000
## [8,] 0.0000000 0.0000000    0    0 0.0000000 0.0000000 0.0000000 0.8548166

```

```

# X
X = matrix(ncol=1, nrow=length(beob), 1); X

```

```

##           [,1]
## [1,]      1
## [2,]      1
## [3,]      1
## [4,]      1
## [5,]      1
## [6,]      1
## [7,]      1
## [8,]      1

```

```

# y
y = as.matrix(data.ped[is.na(data.ped$y) == FALSE, colnames(data.ped) == "y"]); y

```

```

##           [,1]
## [1,]    444
## [2,]    555
## [3,]    550
## [4,]    580
## [5,]    625
## [6,]    375
## [7,]    400
## [8,]    355

```

```

### set up the mixed model equations:
LHS = rbind(cbind((t(X)%*%W.inv%*X), # left hand side
                  (t(X)%*%W.inv%*Z.rhs),
                  (t(X)%*%W.inv%*Z.rhd)),
            cbind((t(Z.rhs)%*%W.inv%*X),
                  (t(Z.rhs)%*%W.inv%*Z.rhs+GAMi*la1),
                  (t(Z.rhs)%*%W.inv%*Z.rhd+GAMi*la2)),
            cbind((t(Z.rhd)%*%W.inv%*X),
                  (t(Z.rhd)%*%W.inv%*Z.rhs+GAMi*la2),
                  (t(Z.rhd)%*%W.inv%*Z.rhd+GAMi*la3)))

RHS = rbind((t(X)%*%W.inv%*y),t(Z.rhs)%*%W.inv%*y,t(Z.rhd)%*%W.inv%*y) # right hand side

### Solving the mixed model equations by direct inversion of the coefficient matrix
### -> solutions in LS
LS = solve(LHS)%*%RHS

#####
##### ASREML: #####
#####

# Now, the results generated in R (saved in vector LS) should be replicated with ASreml.
# For this purpose, the dataset needed for ASreml can be prepared in R.

# The dataset must contain the columns ID = individual; GeV = paternal gamete of parent;
# with phenotype; GeM = maternal gamete of parent with phenotype; GV1 = paternal gamete
# of sire of phenotyped progeny; GV2 = maternal gamete of sire of phenotyped progeny;
# GM1 = paternal gamete of dam of phenotyped progeny; GM2 = maternal gamete of dam of
# phenotyped progeny; y = phenotype; and weight:

setwd("../ProFor")
cross = read.table("crossref.txt", header = FALSE); cross=cross[,c(1:5)]
colnames(cross) = c("IdNew","IdnewSire","IdNewDam","glnr1","glnr2")
pedn = cbind(ped, cross$glnr1, cross$glnr2)
colnames(pedn) = c("id","dad","mom","F","y","IdNew","IdNewSire","IdNewDam","glnr1","glnr2")

## parents with records:
m = match(pedn$id, ind_beob[which(ind_beob$index == 1),1], nomatch = 0)
dat_sub1 = as.data.frame(cbind(pedn[m!=0,colnames(pedn) == "id"],
                              pedn[m!=0,colnames(pedn) == "glnr1"],
                              pedn[m!=0,colnames(pedn) == "glnr2"],
                              rep(0,nrow(pedn[m!=0,])),
                              rep(0,nrow(pedn[m!=0,])),
                              rep(0,nrow(pedn[m!=0,])),
                              rep(0,nrow(pedn[m!=0,])),
                              pedn[m!=0,colnames(pedn) == "y"]))

## animals with records but without offspring
ind_beob2 = ind_beob[-(which(ind_beob$index == 1)),]
ped.sub = pedn[,c(which(colnames(pedn) == "id"),
                  which(colnames(pedn) == "glnr1"),
                  which(colnames(pedn) == "glnr2"))]
dad_sub = merge(ind_beob2, ped.sub, by.x = "dad", by.y = "id")

```

```

mom_sub = merge(dad_sub, ped_sub, by.x = "mom", by.y = "id")
mom_sub = mom_sub[order(mom_sub$ID),]
dat_sub2 = as.data.frame(cbind(mom_sub$ID, rep(0, nrow(mom_sub)),
                                rep(0, nrow(mom_sub)),
                                mom_sub$glnr1.x,
                                mom_sub$glnr2.x,
                                mom_sub$glnr1.y,
                                mom_sub$glnr2.y,
                                mom_sub$y))
datasr = rbind(dat_sub1, dat_sub2)
colnames(datasr) = c("ID", "GeV", "GeM", "GV1", "GV2", "GM1", "GM2", "y")
datasr = datasr[order(datasr$ID),]
datasr = cbind(datasr, (diag(W.inv)))
colnames(datasr)[9] = "weight"

```

```
datasr
```

```

##   ID GeV GeM GV1 GV2 GM1 GM2   y   weight
## 3   6   0   0   3   3   4   4 444 0.8265941
## 4   7   0   0   1   1   4   4 555 0.8265941
## 1   8   6   7   0   0   0   0 550 1.0000000
## 2  10   9  10   0   0   0   0 580 1.0000000
## 5  11   0   0   9  10   8   8 625 0.8265941
## 6  12   0   0   5   5   6   7 375 0.8548166
## 7  13   0   0   9  10   8   8 400 0.8265941
## 8  14   0   0   5   5   6   7 355 0.8548166

```

```

# write dataset (datasr) to ASreml directory:
setwd("../ASreml")
write.table(datasr, "dataset.txt", col.names=TRUE, row.names=FALSE, quote=F, sep="\t")

```

```
### run ASreml from R-script using the system function:
```

```

#####
#
# system("asreml -l command_file.as") #
#
#####

```

```
#####
```

```

### Now, compare results of R and ASreml:
ASR = read.table("command_file.sln", header = FALSE)
result=as.data.frame(cbind(ASR$V3,round(LS,3)))
colnames(result)=c("ASreml_solution", "R_solution")
result

```

```

##   ASreml_solution R_solution
## 1          493.400      493.440
## 2          -3.144      -3.144
## 3           9.327       9.327
## 4          -7.183      -7.183
## 5          -5.235      -5.235
## 6         -11.550     -11.546
## 7          -5.910      -5.910

```

```

## 8          -7.675      -7.675
## 9           1.967       1.967
## 10         -4.460      -4.460
## 11         17.760      17.760
## 12         -3.312      -3.312
## 13          9.779       9.779
## 14         -7.525      -7.525
## 15         -5.488      -5.488
## 16        -12.090     -12.090
## 17         -6.230      -6.230
## 18         -8.046      -8.046
## 19          2.066       2.066
## 20         -4.677      -4.677
## 21         18.620      18.620

#####
# use the solutions to calculate the breeding values and imprinting effects:
maxzahlgl = max(max(cross$glnr1),max(cross$glnr2))

LSi = LS[-1]

LSasFather = LSi[1:(maxzahlgl)]
LSasMother = LSi[(maxzahlgl+1):(maxzahlgl*2)]

effects = cbind(LSasFather,LSasMother) # effects (TAs,GAM) as father and as mother
effkinds = c("TA","TA","TA","TA","TA","GAM","GAM","TA","GAM","GAM")
effects = as.data.frame(cbind(effects, effkinds))
effects$effkinds = as.character(effects$effkinds)

## Calculation of breeding values and imprinting effects (EIE):
BV = as.data.frame(matrix(nrow=nrow(pedn), ncol=2, NA))
colnames(BV) = c("BVasV","BVasM")
i = 0
j = 1
for(i in 1:nrow(BV)){
  if(effects[j,colnames(effects) == "effkinds"] == "TA"){
    BV[i,colnames(BV) == "BVasV"] =
      as.numeric(as.character(effects[j,colnames(effects) == "LSasFather"]))*2
    BV[i,colnames(BV) == "BVasM"] =
      as.numeric(as.character(effects[j,colnames(effects) == "LSasMother"]))*2
  }
  if(effects[j,colnames(effects)=="effkinds"] == "GAM"){
    BV[i,colnames(BV) == "BVasV"] =
      as.numeric(as.character(effects[j,colnames(effects) == "LSasFather"])) +
      as.numeric(as.character(effects[j+1,colnames(effects) == "LSasFather"]))
    BV[i,colnames(BV) == "BVasM"] =
      as.numeric(as.character(effects[j,colnames(effects) == "LSasMother"])) +
      as.numeric(as.character(effects[j+1,colnames(effects) == "LSasMother"]))
    j = j + 1
  }
  j = j + 1
}

## adding the imprinting effect:

```

```
BV$EIE = BV$BVasM - BV$BVasV
BV
```

```
##      BVasV      BVasM      EIE
## 1 -6.288844 -6.624458 -0.3356146
## 2 18.654704 19.558793  0.9040888
## 3 -14.366972 -15.050400 -0.6834284
## 4 -10.470649 -10.975545 -0.5048968
## 5 -23.091491 -24.179139 -1.0876486
## 6 -13.584288 -14.275236 -0.6909483
## 7  3.933645  4.131118  0.1974730
## 8 13.299464 13.943774  0.6443093
```

```
# BVasV = breeding value as sire
# BVasM = breeding value as dam
# EIE = imprinting effect
setwd("../../../")
```

Toy example - the gametic model with maternal genetic effect

Inga Blunk

1 Apr 2020

This script provides an example for the *gametic model* including a *maternal genetic effect*. In matrix notation the model can be written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{gs}\mathbf{g}_s + \mathbf{Z}_{gd}\mathbf{g}_d + \mathbf{Z}_m\mathbf{m} + \mathbf{e},$$

where $\mathbf{g}_s$, $\mathbf{g}_d$, and $\mathbf{m}$ are vectors containing gametic effects under a paternal expression pattern, gametic effects under a maternal expression pattern, and gametic maternal genetic effects. The matrices $\mathbf{Z}_{gs}$, $\mathbf{Z}_{gd}$ connect the observations with the expressed parental gametes. Matrix $\mathbf{Z}_m$ connects observations with the gametes of the dam of the individuals with records. The gametic variances are:

$$\text{Var} \begin{bmatrix} \mathbf{g}_s \\ \mathbf{g}_d \\ \mathbf{m} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{G}\sigma_s^2 & \mathbf{G}\sigma_{sd} & \mathbf{G}\sigma_{sm} & 0 \\ \mathbf{G}\sigma_{sd} & \mathbf{G}\sigma_d^2 & \mathbf{G}\sigma_{dm} & 0 \\ \mathbf{G}\sigma_{sm} & \mathbf{G}\sigma_{dm} & \mathbf{G}\sigma_m^2 & 0 \\ 0 & 0 & 0 & \mathbf{I}\sigma_e^2 \end{bmatrix}.$$

$\mathbf{G}$ is the gametic relationship matrix (Schaeffer *et al.*, 1989). The gametic variances are used in the mixed model equations as:

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_2 & \lambda_4 & \lambda_5 \\ \lambda_3 & \lambda_5 & \lambda_6 \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & \sigma_{sd} & \sigma_{sm} \\ \sigma_{sd} & \sigma_d^2 & \sigma_{dm} \\ \sigma_{sm} & \sigma_{dm} & \sigma_m^2 \end{bmatrix}^{-1} \sigma_e^2.$$

The mixed model equations are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z}_{gs} & \mathbf{X}'\mathbf{Z}_{gd} & \mathbf{X}'\mathbf{Z}_m \\ \mathbf{Z}_{gs}'\mathbf{X} & \mathbf{Z}_{gs}'\mathbf{Z}_{gs} + \mathbf{G}^{-1}\lambda_1 & \mathbf{Z}_{gs}'\mathbf{Z}_{gd} + \mathbf{G}^{-1}\lambda_2 & \mathbf{Z}_{gs}'\mathbf{Z}_m + \mathbf{G}^{-1}\lambda_3 \\ \mathbf{Z}_{gd}'\mathbf{X} & \mathbf{Z}_{gd}'\mathbf{Z}_{gs} + \mathbf{G}^{-1}\lambda_2 & \mathbf{Z}_{gd}'\mathbf{Z}_{gd} + \mathbf{G}^{-1}\lambda_4 & \mathbf{Z}_{gd}'\mathbf{Z}_m + \mathbf{G}^{-1}\lambda_5 \\ \mathbf{Z}_m'\mathbf{X} & \mathbf{Z}_m'\mathbf{Z}_{gs} + \mathbf{G}^{-1}\lambda_3 & \mathbf{Z}_m'\mathbf{Z}_{gd} + \mathbf{G}^{-1}\lambda_5 & \mathbf{Z}_m'\mathbf{Z}_m + \mathbf{G}^{-1}\lambda_6 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{g}}_s \\ \hat{\mathbf{g}}_d \\ \hat{\mathbf{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}_{gs}'\mathbf{y} \\ \mathbf{Z}_{gd}'\mathbf{y} \\ \mathbf{Z}_m'\mathbf{y} \end{bmatrix}.$$

Solving the mixed model equations by direct inversion of the coefficient matrix, provides $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{g}}_s$, $\hat{\mathbf{g}}_d$, and $\hat{\mathbf{m}}$.

References

- Gilmour, A.R., Gogel, B.J., Cullis, B.R., and Thompson R. 2009. ASReml user guide release 3.0. VSN International Ltd., Hemel Hempstead, UK. <https://asreml.kb.vsn.co.uk/wp-content/uploads/sites/3/2018/02/ASReml-3-User-Guide.pdf>
- Schaeffer, L.R., Kennedy, B.W., and Gibson, J.P. 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Example

In the following a toy example is provided in an R environment, i.e. based on a toy dataset the mixed model equations of the *gametic model* with a *maternal genetic effect* will be solved in R. Subsequently, the same dataset will be analyzed with ASReml to show how this model can be implemented into the software. Eventually, the R and ASReml solutions will be compared. At the end of the R script breeding values, imprinting effects, and maternal genetic effects will be calculated. They can be compared to the results generated with the other corresponding *gametic models* in order to show their equivalences. The fortran program *GGRInv* that builds the inverse gametic relationship matrix $\mathbf{G}$ and the ASReml software (version 3.0; Gilmour *et al.*, 2009) can be navigated from the following R script. The corresponding command files

(command_file.txt for *GGRinv*; command_file.as for *ASreml*) must be available within the corresponding directories, which are *ProFor* and *ASreml*, respectively. The example was build in an UNIX environment.

```
### Overall number of animals in the pedigree:
```

```
nr_animals = 14
```

```
### data file:
```

```
# F = inbreeding coefficient; y = phenotype
```

```
data.ped = data.frame(id = seq(1,nr_animals,1),
  dad = c(0,0,1,1,3,3,1,1,3,5,10,5,10,5),
  mom = c(0,0,0,0,4,4,4,4,2,2,9,8,9,8),
  F = c(0,0,0,0,0.125,0.125,0.250,0.250,0,0,0.203125,0.28125,0.203125,0.28125),
  y = c(NA,NA,NA,NA,NA,444,555,550,NA,580,625,375,400,355))
```

```
data.ped # toy data to be analyzed in the following:
```

##	id	dad	mom	F	y
## 1	1	0	0	0.000000	NA
## 2	2	0	0	0.000000	NA
## 3	3	1	0	0.000000	NA
## 4	4	1	0	0.000000	NA
## 5	5	3	4	0.125000	NA
## 6	6	3	4	0.125000	444
## 7	7	1	4	0.250000	555
## 8	8	1	4	0.250000	550
## 9	9	3	2	0.000000	NA
## 10	10	5	2	0.000000	580
## 11	11	10	9	0.203125	625
## 12	12	5	8	0.281250	375
## 13	13	10	9	0.203125	400
## 14	14	5	8	0.281250	355

```
#
```

```
### gametic variance components:
```

```
sigma.s.2 = 2552      # gametic variance as father
sigma.sd = 2670       # covariance between gametic effects as father and mother
sigma.sm = 962        # covariance between gametic effect as father and maternal genetic effect
sigma.d.2 = 2800      # gametic variances as mother
sigma.dm = 892        # covariance between gametic effect as mother and maternal genetic effect
sigma.m.2 = 1989      # gametic maternal variance
sigma.e.2 = 12756     # residual variance in a gametic model
```

```
### construct lambdas for the coefficient matrix (left hand side of mixed model equations):
```

```
VAR = matrix(nrow=3, ncol=3, 0)
VAR[1,1] = sigma.s.2; VAR[1,2] = sigma.sd; VAR[1,3] = sigma.sm
VAR[2,1] = sigma.sd; VAR[2,2] = sigma.d.2; VAR[2,3] = sigma.dm
VAR[3,1] = sigma.sm; VAR[3,2] = sigma.dm; VAR[3,3] = sigma.m.2
```

```
# inverse:
```

```
VARi = solve(VAR)
```

```

# calculate lambdas:
lambda = VARi*sigma.e.2
la1 = lambda[1,1]; la2 = lambda[1,2]; la3 = lambda[1,3]
      la4 = lambda[2,2]; la5 = lambda[2,3]
      la6 = lambda[3,3]

### construction of the inverse of the gametic relationship matrix with the fortran
### program "GGRInv" (as 3-column lower triangle):
setwd("ProFor") # change to the ProFor directory containing "GGRInv"

# prepare the dataset needed for "GGRInv". The dataset must provide the following
# columns: "id","dad","mom","indicator","F"
data_in = cbind(data.ped[,1:3],rep(2,nrow(data.ped)),data.ped$F)
colnames(data_in)[c(4,5)] = c("indicator","F")

data_in # note: the indicator vector of 2s -> construction of full gametic relationship matrix G

##      id dad mom indicator      F
## 1      1   0   0          2 0.000000
## 2      2   0   0          2 0.000000
## 3      3   1   0          2 0.000000
## 4      4   1   0          2 0.000000
## 5      5   3   4          2 0.125000
## 6      6   3   4          2 0.125000
## 7      7   1   4          2 0.250000
## 8      8   1   4          2 0.250000
## 9      9   3   2          2 0.000000
## 10     10   5   2          2 0.000000
## 11     11  10   9          2 0.203125
## 12     12   5   8          2 0.281250
## 13     13  10   9          2 0.203125
## 14     14   5   8          2 0.281250

# write dataset to ProFor directory:
write.table(data_in, "pedigree_input.txt", col.names=FALSE, row.names=FALSE, quote=FALSE, sep="\t")

# run "GGRInv" to construct the inverse of G (AImatrix.giv) with system function:

#####
#
# system("GGRInv > out.txt") #
#
#####

# move inverse of G (AImatrix.giv) to the ASreml directory so it will be available for the
# subsequent analysis with the ASreml software:

# system("mv AImatrix.giv ../ASreml")

### rewrite inverse gametic relationship matrix to a n x n matrix:
setwd("../ASreml") # change to the ASreml directory
GAMinv = read.table("AImatrix.giv", header=FALSE) # 3-column lower triangle of inverse of G
GAMi = matrix(ncol=nr_animals*2, nrow=nr_animals*2, 0) # inverse of G as n x n matrix
i = 0

```

```

for(i in 1:nrow(GAMinv)){
  GAMi[ GAMinv[i,1] , GAMinv[i,2]] = GAMinv[i,3]
  GAMi[ GAMinv[i,2], GAMinv[i,1]] = GAMinv[i,3]
}
GAMi[1:10,1:10]

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]      [,9]      [,10]
## [1,]    3    2    0    0 -1.0  0.0  -1    0  0.000000  0.000000
## [2,]    2    3    0    0 -1.0  0.0  -1    0  0.000000  0.000000
## [3,]    0    0    2    1  0.0  0.0    0    0  0.000000  0.000000
## [4,]    0    0    1    2  0.0  0.0    0    0  0.000000  0.000000
## [5,]   -1   -1    0    0  3.5  1.5    0    0 -1.000000  0.000000
## [6,]    0    0    0    0  1.5  2.5    0    0 -1.000000  0.000000
## [7,]   -1   -1    0    0  0.0  0.0    4    2  0.000000 -1.000000
## [8,]    0    0    0    0  0.0  0.0    2    3  0.000000 -1.000000
## [9,]    0    0    0    0 -1.0 -1.0    0    0  3.714286  1.714286
## [10,]   0    0    0    0  0.0  0.0   -1   -1  1.714286  3.714286

### construction of incidence matrices to set up the mixed model equations:
beob = data.ped[is.na(data.ped$y) == FALSE,1] # IDs of animals with phenotypes

## construct Z.gs
index_p = matrix(nrow = nr_animals, ncol = 2,0); index_p[,1] = seq(1,nr_animals); l = 0; la = 0
for(k in 1:nr_animals){
  if(k == 1){l = 1; la = 1}else{l = k+la ; la = la+1}
  index_p[k,2] = l
}
Z.gs = matrix(ncol = (nr_animals)*2, nrow = length(beob),0)
for(i in 1:length(which(is.na(data.ped$y)==FALSE))){
  tier=beob[i]
  ii = index_p[index_p[,1] == tier,2]
  Z.gs[i,ii] = 1
}
Z.gs # note that the records of animals are linked to their expressed paternal gametes:

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    0    0    0    0    0    1    0    0
## [2,]    0    0    0    0    0    0    0    0    0    0    0    0    1
## [3,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [4,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,]    0    0    0    0    0    0    0    0    0    0    0
## [2,]    0    0    0    0    0    0    0    0    0    0    0
## [3,]    0    1    0    0    0    0    0    0    0    0    0
## [4,]    0    0    0    0    0    1    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    1    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    1    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0
##      [,25] [,26] [,27] [,28]

```

```

## [1,] 0 0 0 0
## [2,] 0 0 0 0
## [3,] 0 0 0 0
## [4,] 0 0 0 0
## [5,] 0 0 0 0
## [6,] 0 0 0 0
## [7,] 1 0 0 0
## [8,] 0 0 1 0

## construct Z.gd
index_m = matrix(nrow = nr_animals, ncol = 2, 0); index_m[,1] = seq(1,nr_animals); l = 0; la = 0
for(k in 1:nr_animals){
  if(k == 1){l = 2; la = 2}else{l = k+la ; la = la+1}
  index_m[k,2] = 1
}
Z.gd = matrix(ncol = (nr_animals)*2, nrow = length(beob),0)
for(i in 1:length(beob)){
  tier = beob[i]
  ii = index_m[index_m[,1] == tier,2]
  Z.gd[i,ii] = 1
}

Z.gd # note that the records of animals are linked to their expressed maternal gametes:

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0 1 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,] 0 0 0 0 0 0 0 0 0 0 0
## [2,] 1 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 1 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 1 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 1 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 1
## [7,] 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0
##      [,25] [,26] [,27] [,28]
## [1,] 0 0 0 0
## [2,] 0 0 0 0
## [3,] 0 0 0 0
## [4,] 0 0 0 0
## [5,] 0 0 0 0
## [6,] 0 0 0 0
## [7,] 0 1 0 0
## [8,] 0 0 0 1

# construct Z.m:
Z.m = matrix(ncol = ncol(GAMi), nrow = length(beob), 0)

```

```

# to construct Z.m the gametic effect numbers of the dams of the individuals with records
# are needed. These numbers are provided in column 8 and 9 of the cross-reference table
# (crossref.txt) which was generated with GGRinv when the inverse of G was constructed:
cross=read.table("../ProFor/crossref.txt", header=FALSE)
colnames(cross)=c("ID","dad","mom","GoF","GoM","GfF","GfM","GmF","GmM")
crossY=cross[is.na(data.ped$y)==FALSE,]
for(i in 1:nrow(crossY)){
  Z.m[i,crossY[i,c(which(colnames(crossY=="GmF"),which(colnames(crossY=="GmM")))[1,1]] = 1
  Z.m[i,crossY[i,c(which(colnames(crossY=="GmF"),which(colnames(crossY=="GmM")))[1,2]] = 1
}
Z.m # note that the records of animals are linked to both gametes of their dams:

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    0    1    1    0    0    0    0    0
## [2,]    0    0    0    0    0    0    1    1    0    0    0    0    0
## [3,]    0    0    0    0    0    0    1    1    0    0    0    0    0
## [4,]    0    0    1    1    0    0    0    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,]      0      0      0      0      0      0      0      0      0      0      0
## [2,]      0      0      0      0      0      0      0      0      0      0      0
## [3,]      0      0      0      0      0      0      0      0      0      0      0
## [4,]      0      0      0      0      0      0      0      0      0      0      0
## [5,]      0      0      0      1      1      0      0      0      0      0      0
## [6,]      0      1      1      0      0      0      0      0      0      0      0
## [7,]      0      0      0      1      1      0      0      0      0      0      0
## [8,]      0      1      1      0      0      0      0      0      0      0      0
##      [,25] [,26] [,27] [,28]
## [1,]      0      0      0      0
## [2,]      0      0      0      0
## [3,]      0      0      0      0
## [4,]      0      0      0      0
## [5,]      0      0      0      0
## [6,]      0      0      0      0
## [7,]      0      0      0      0
## [8,]      0      0      0      0

```

```

# X: the only fixed effect is the population mean:
X = matrix(ncol = 1, nrow = length(beob),1);X

```

```

##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
## [4,]    1
## [5,]    1
## [6,]    1
## [7,]    1
## [8,]    1

```

```

## y
y = as.matrix(data.ped[is.na(data.ped$y) == FALSE,colnames(data.ped) == "y"])

```

y

```
##      [,1]
## [1,]  444
## [2,]  555
## [3,]  550
## [4,]  580
## [5,]  625
## [6,]  375
## [7,]  400
## [8,]  355
```

set up the mixed model equations:

```
LHS = rbind(cbind((t(X)%*%X), # left hand side
                  (t(X)%*%Z.gs),
                  (t(X)%*%Z.gd),
                  (t(X)%*%Z.m)),
            cbind((t(Z.gs)%*%X),
                  (t(Z.gs)%*%Z.gs+GAMi*la1),
                  (t(Z.gs)%*%Z.gd+GAMi*la2),
                  (t(Z.gs)%*%Z.m+GAMi*la3)),
            cbind((t(Z.gd)%*%X),
                  (t(Z.gd)%*%Z.gs+GAMi*la2),
                  (t(Z.gd)%*%Z.gd+GAMi*la4),
                  (t(Z.gd)%*%Z.m+GAMi*la5)),
            cbind((t(Z.m)%*%X),
                  (t(Z.m)%*%Z.gs+GAMi*la3),
                  (t(Z.m)%*%Z.gd+GAMi*la5),
                  (t(Z.m)%*%Z.m+GAMi*la6)))
```

```
RHS = rbind((t(X)%*%y),t(Z.gs)%*%y,t(Z.gd)%*%y,t(Z.m)%*%y) # right hand side
```

Solving the mixed model equations by direct inversion of the coefficient matrix

-> solutions in LS

```
LS = solve(LHS)%*%RHS
```

```
#####
##### ASREML: #####
#####
```

*# Now, the results generated in R (saved in vector LS) should be replicated with ASreml.
For this purpose, the dataset needed for ASreml can be prepared in R.*

*# The dataset must contain the columns ID = individual; g1nr = paternal gamete; g2nr =
maternal gamete; mor1 = paternal gamete of mother; mor2 = maternal gamete of mother;
and the phenotype. The cross-reference table already provides all variables except for
the phenotype:*

```
cross=cross[,c(which(colnames(cross)== "ID"),
               which(colnames(cross)== "GoF"),
               which(colnames(cross)== "GoM"),
               which(colnames(cross)== "GmF"),
               which(colnames(cross)== "GmM"))]
```

```
PPneu=as.data.frame(cbind(cross,data.ped[,colnames(data.ped)=="y"])) # add phenotype
```

```

colnames(PPneu)=c("ID","g1nr","g2nr","mor1","mor2","phenotype")
# write dataset (PPneu) to ASreml directory:
write.table(PPneu,"dataset.txt",col.names=TRUE, row.names=FALSE, sep="\t", quote=FALSE)

### run ASreml from R-script using the system function:

#####
#
# system("asreml -l command_file.as") #
#
#####

#####

### Now, compare results of R and ASreml:
ASR = read.table("command_file.sln", header=FALSE) # read in the ASreml solution file

results=as.data.frame(cbind(ASR[,3],round(LS,3)))
colnames(results)=c("ASreml_solution","R_solution")
results

```

```

##      ASreml_solution R_solution
## 1      502.9000      502.915
## 2      -4.7300      -4.730
## 3      -4.7300      -4.730
## 4       9.1720       9.172
## 5       9.1720       9.172
## 6      -8.6710      -8.671
## 7      -7.8810      -7.881
## 8      -5.2320      -5.232
## 9      -1.0030      -1.003
## 10     -12.9500     -12.946
## 11      -7.7880      -7.788
## 12     -11.7000     -11.699
## 13      -6.7000      -6.700
## 14       0.7309       0.731
## 15       2.5960       2.596
## 16     -10.4800     -10.480
## 17      -8.5350      -8.535
## 18      -8.0620      -8.062
## 19       9.3850       9.385
## 20      -5.9660      -5.966
## 21      14.4300      14.429
## 22      13.6500      13.652
## 23      10.5200      10.518
## 24     -15.9000     -15.905
## 25     -14.4700     -14.474
## 26      -4.9520      -4.952
## 27      -8.9470      -8.947
## 28     -17.4000     -17.401
## 29     -15.8200     -15.816
## 30      -4.4170      -4.417
## 31      -4.4170      -4.417
## 32       9.1580       9.158

```

## 33	9.1580	9.158
## 34	-8.5440	-8.544
## 35	-8.2550	-8.255
## 36	-5.0300	-5.030
## 37	-1.2270	-1.227
## 38	-13.2900	-13.286
## 39	-8.0150	-8.015
## 40	-11.9800	-11.982
## 41	-6.8850	-6.885
## 42	1.2970	1.297
## 43	2.8630	2.863
## 44	-9.8060	-9.806
## 45	-8.1520	-8.152
## 46	-8.1870	-8.187
## 47	9.3710	9.371
## 48	-6.0460	-6.046
## 49	14.6700	14.670
## 50	14.1700	14.168
## 51	10.9300	10.928
## 52	-16.4400	-16.445
## 53	-14.1900	-14.187
## 54	-5.2960	-5.296
## 55	-9.4840	-9.484
## 56	-18.0100	-18.010
## 57	-15.5900	-15.595
## 58	-9.2930	-9.293
## 59	-9.2930	-9.293
## 60	9.6420	9.642
## 61	9.6420	9.642
## 62	-10.7100	-10.706
## 63	-2.8250	-2.825
## 64	-8.2300	-8.230
## 65	2.1260	2.126
## 66	-8.5260	-8.526
## 67	-4.8120	-4.812
## 68	-8.0560	-8.056
## 69	-4.2480	-4.248
## 70	-7.2350	-7.235
## 71	-1.1430	-1.143
## 72	-20.3000	-20.296
## 73	-14.2500	-14.251
## 74	-6.5390	-6.539
## 75	9.8680	9.868
## 76	-5.0100	-5.010
## 77	11.4000	11.404
## 78	6.7480	6.748
## 79	4.9570	4.957
## 80	-8.7570	-8.757
## 81	-18.9300	-18.933
## 82	-0.2651	-0.265
## 83	-1.5450	-1.545
## 84	-9.3210	-9.321
## 85	-19.3800	-19.381

```
#####
# use solutions to calculate the breeding values, maternal effects and imprinting effects:
LSi=LS[-1]
LSasFather=LSi[1:(nr_animals*2)]
LSasMother=LSi[(nr_animals*2+1):(nr_animals*4)]
LSmaternal=LSi[(nr_animals*4+1):length(LSi)]
LSasFathergv=LSasFather[seq(1,(nr_animals*2),2)]
LSasFathergm=LSasFather[seq(2,(nr_animals*2),2)]
asV=cbind(LSasFathergv,LSasFathergm)
asV # gametic effects as sire:
```

```
##      LSasFathergv LSasFathergm
## [1,]    -4.7302531    -4.730253
## [2,]     9.1722660     9.172266
## [3,]    -8.6705507    -7.880595
## [4,]    -5.2319429    -1.003380
## [5,]   -12.9455305    -7.787619
## [6,]   -11.6993809    -6.699780
## [7,]     0.7308645     2.595969
## [8,]   -10.4796366    -8.534860
## [9,]    -8.0624026     9.385436
## [10,]   -5.9657810    14.428823
## [11,]    13.6520380    10.517622
## [12,]   -15.9049902   -14.473960
## [13,]    -4.9521611    -8.946802
## [14,]   -17.4013885   -15.815893
```

```
LSasMothergv=LSasMother[seq(1,(nr_animals*2),2)]
LSasMothergm=LSasMother[seq(2,(nr_animals*2),2)]
asM=cbind(LSasMothergv,LSasMothergm)
asM # gametic effects as dam:
```

```
##      LSasMothergv LSasMothergm
## [1,]    -4.416712    -4.416712
## [2,]     9.158033     9.158033
## [3,]    -8.544362    -8.255301
## [4,]    -5.030358    -1.227292
## [5,]   -13.285720    -8.014713
## [6,]   -11.981950    -6.885354
## [7,]     1.296918     2.862997
## [8,]    -9.805757    -8.151904
## [9,]    -8.187125     9.370740
## [10,]   -6.045938    14.670237
## [11,]    14.168255    10.927798
## [12,]   -16.444718   -14.187367
## [13,]    -5.296170    -9.484333
## [14,]   -18.010308   -15.594638
```

```
LSmaternalGV=LSmaternal[seq(1,(nr_animals*2),2)]
LSmaternalGM=LSmaternal[seq(2,(nr_animals*2),2)]
MATg=cbind(LSmaternalGV,LSmaternalGM)
MATg # gametic maternal genetic effects:
```

```
##      LSmaternalGV LSmaternalGM
```

```
## [1,] -9.2931206 -9.293121
## [2,] 9.6424274 9.642427
## [3,] -10.7056411 -2.825041
## [4,] -8.2298810 2.126479
## [5,] -8.5257234 -4.812083
## [6,] -8.0559772 -4.248424
## [7,] -7.2345018 -1.142878
## [8,] -20.2955790 -14.251218
## [9,] -6.5393639 9.868405
## [10,] -5.0099835 11.403628
## [11,] 6.7479732 4.957272
## [12,] -8.7566603 -18.932684
## [13,] -0.2650516 -1.545450
## [14,] -9.3207415 -19.381000
```

```
### breeding values, maternal genetic effects and imprinting effects:
```

```
BVasV=asV[,1]+asV[,2] # breeding value as sire
BVasM=asM[,1]+asM[,2] # breeding value as dam
MatGenEff=MATg[,1]+MATg[,2] # mat. genetic effect
EIE = BVasM - BVasV # imprinting effects
```

```
cbind(BVasV,BVasM,MatGenEff,EIE)
```

```
##          BVasV      BVasM  MatGenEff      EIE
## [1,] -9.460506 -8.833423 -18.586241  0.62708288
## [2,] 18.344532 18.316067 19.284855 -0.02846512
## [3,] -16.551146 -16.799663 -13.530682 -0.24851699
## [4,] -6.235323 -6.257649 -6.103402 -0.02232668
## [5,] -20.733149 -21.300433 -13.337807 -0.56728331
## [6,] -18.399161 -18.867304 -12.304401 -0.46814298
## [7,] 3.326833 4.159916 -8.377379 0.83308233
## [8,] -19.014497 -17.957661 -34.546797 1.05683520
## [9,] 1.323034 1.183615 3.329041 -0.13941910
## [10,] 8.463042 8.624299 6.393644 0.16125774
## [11,] 24.169660 25.096053 11.705245 0.92639278
## [12,] -30.378950 -30.632086 -27.689344 -0.25313594
## [13,] -13.898963 -14.780502 -1.810502 -0.88153879
## [14,] -33.217281 -33.604946 -28.701742 -0.38766434
```

```
setwd("?.")
```

Toy example - the generalized gametic model with maternal gametic effect

Inga Blunk

1 Apr 2020

This script provides an example for the *generalized gametic model* including a *maternal genetic effect*. In matrix notation the model is:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{hs}\mathbf{a}_s + \mathbf{Z}_{hd}\mathbf{a}_d + \mathbf{Z}_m\mathbf{m} + \mathbf{e},$$

where $\mathbf{y}$ is a vector of observations, $\boldsymbol{\beta}$ comprises fixed effects and $\mathbf{X}$ is the corresponding incidence matrix. The covariance of random effects is assumed to be:

$$\text{Var} \begin{bmatrix} \mathbf{a}_s \\ \mathbf{a}_d \\ \mathbf{m} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{G}}\sigma_s^2 & \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_{sm} & 0 \\ \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_d^2 & \bar{\mathbf{G}}\sigma_{dm} & 0 \\ \bar{\mathbf{G}}\sigma_{sm} & \bar{\mathbf{G}}\sigma_{dm} & \bar{\mathbf{G}}\sigma_m^2 & 0 \\ 0 & 0 & 0 & \mathbf{I}\sigma_e^2 \end{bmatrix}.$$

The *generalized gametic model* includes, apart from the maternal genetic effect $\mathbf{m}$, the genetic effect vectors $\mathbf{a}_s$ and $\mathbf{a}_d$ which contain gametic effects for phenotyped individuals and transmitting abilities for animals without own observations. Consequently, the model uses the generalized gametic relationship matrix $\bar{\mathbf{G}}$ instead of the classical gametic relationship matrix $\mathbf{G}$ (Schaeffer *et al.*, 1989). Further, the incidence matrices $\mathbf{Z}_{hs}$ and $\mathbf{Z}_{hd}$ link observations to the random gametic effects in $\mathbf{a}_s$ and $\mathbf{a}_d$, while no observation is linked to any of the transmitting abilities in the latter vectors. Incidence matrix $\mathbf{Z}_m$ link observations to the genetic effects of mothers. In case the mother does not have an observation on her own, an entry of 2 is needed at the position of her transmitting ability. In case, the mother has a phenotype a 1 is needed at the position of her two gametes, respectively. In this way the *gametic* maternal genetic variance is estimated. The gametic variances are used in the mixed model equations as:

$$\begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_2 & \lambda_4 & \lambda_5 \\ \lambda_3 & \lambda_5 & \lambda_6 \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & \sigma_{sd} & \sigma_{sm} \\ \sigma_{sd} & \sigma_d^2 & \sigma_{dm} \\ \sigma_{sm} & \sigma_{dm} & \sigma_m^2 \end{bmatrix}^{-1} \sigma_e^2.$$

The mixed model equations are:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z}_{hs} & \mathbf{X}'\mathbf{Z}_{hd} & \mathbf{X}'\mathbf{Z}_m \\ \mathbf{Z}_{hs}'\mathbf{X} & \mathbf{Z}_{hs}'\mathbf{Z}_{hs} + \bar{\mathbf{G}}^{-1}\lambda_1 & \mathbf{Z}_{hs}'\mathbf{Z}_{hd} + \bar{\mathbf{G}}^{-1}\lambda_2 & \mathbf{Z}_{hs}'\mathbf{Z}_m + \bar{\mathbf{G}}^{-1}\lambda_3 \\ \mathbf{Z}_{hd}'\mathbf{X} & \mathbf{Z}_{hd}'\mathbf{Z}_{hs} + \bar{\mathbf{G}}^{-1}\lambda_2 & \mathbf{Z}_{hd}'\mathbf{Z}_{hd} + \bar{\mathbf{G}}^{-1}\lambda_4 & \mathbf{Z}_{hd}'\mathbf{Z}_m + \bar{\mathbf{G}}^{-1}\lambda_5 \\ \mathbf{Z}_m'\mathbf{X} & \mathbf{Z}_m'\mathbf{Z}_{hs} + \bar{\mathbf{G}}^{-1}\lambda_3 & \mathbf{Z}_m'\mathbf{Z}_{hd} + \bar{\mathbf{G}}^{-1}\lambda_5 & \mathbf{Z}_m'\mathbf{Z}_m + \bar{\mathbf{G}}^{-1}\lambda_6 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{a}}_s \\ \hat{\mathbf{a}}_d \\ \hat{\mathbf{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}_{hs}'\mathbf{y} \\ \mathbf{Z}_{hd}'\mathbf{y} \\ \mathbf{Z}_m'\mathbf{y} \end{bmatrix}.$$

Solving the mixed model equations by direct inversion of the coefficient matrix, provides $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{a}}_s$, $\hat{\mathbf{a}}_d$, and $\hat{\mathbf{m}}$.

References

- Gilmour, A.R., Gogel, B.J., Cullis, B.R., and Thompson R. 2009. ASReml user guide release 3.0. VSN International Ltd., Hemel Hempstead, UK. <https://asreml.kb.vsnl.co.uk/wp-content/uploads/sites/3/2018/02/ASReml-3-User-Guide.pdf>
- Schaeffer, L.R., Kennedy, B.W., and Gibson, J.P. 1989. The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266–1272.

Example

In the following a toy example is provided in an R environment, i.e. based on a toy dataset the mixed model equations of the *generalized gametic model* with a *maternal genetic effect* will be solved in R. Subsequently, the same dataset will be analyzed with ASreml to show how this model can be implemented into the software. Eventually, the R and ASreml solutions will be compared. At the end of the R script breeding values, imprinting effects, and maternal genetic effects will be calculated. They can be compared to the results generated with the other corresponding *gametic models* in order to show their equivalences. The fortran program *GGRinv* that builds the inverse gametic relationship matrix $\mathbf{G}$ and the ASreml software (version 3.0; Gilmour *et al.*, 2009) can be navigated from the following R script. The corresponding command files (command_file.txt for *GGRinv*; command_file.as for ASreml) must be available within the corresponding directories, which are *ProFor* and *ASreml*, respectively. The example was build in an UNIX environment.

```
### Overall number of animals in the pedigree:

nr_animals = 14

### data file:
# F = inbreeding coefficient; y = phenotype
data.ped = data.frame(id = seq(1,nr_animals,1),
                      dad = c(0,0,1,1,3,3,1,1,3,5,10,5,10,5),
                      mom = c(0,0,0,0,4,4,4,4,2,2,9,8,9,8),
                      F = c(0,0,0,0,0.125,0.125,0.250,0.250,0,0,0.203125,0.28125,0.203125,0.28125),
                      y = c(NA,NA,NA,NA,NA,NA,444,555,550,NA,580,625,375,400,355))

beob = data.ped[is.na(data.ped$y)==FALSE,1]

data.ped # toy data to be analyzed in the following:

##      id dad mom      F      y
## 1     1   0   0 0.000000    NA
## 2     2   0   0 0.000000    NA
## 3     3   1   0 0.000000    NA
## 4     4   1   0 0.000000    NA
## 5     5   3   4 0.125000    NA
## 6     6   3   4 0.125000  444
## 7     7   1   4 0.250000  555
## 8     8   1   4 0.250000  550
## 9     9   3   2 0.000000    NA
## 10    10   5   2 0.000000  580
## 11    11  10   9 0.203125  625
## 12    12   5   8 0.281250  375
## 13    13  10   9 0.203125  400
## 14    14   5   8 0.281250  355

### gametic variance components:
sigma.s.2 = 2552      # gametic variance as father
sigma.sd  = 2670      # covariance between gametic effects as father and mother
sigma.sm  = 962       # covariance between gametic effect as father and maternal genetic effect
sigma.d.2 = 2800      # gametic variances as mother
sigma.dm  = 892       # covariance between gametic effect as mother and maternal genetic effect
sigma.m.2 = 1989      # maternal gametic variance
sigma.e.2 = 12756     # residual variance in a gametic model

### construct lambdas for the coefficient matrix (left hand side of mixed model equations):
```

```

VAR=matrix(nrow=3, ncol=3, 0)
VAR[1,1] = sigma.s.2; VAR[1,2] = sigma.sd; VAR[1,3] = sigma.sm
VAR[2,1] = sigma.sd; VAR[2,2] = sigma.d.2; VAR[2,3] = sigma.dm
VAR[3,1] = sigma.sm; VAR[3,2] = sigma.dm; VAR[3,3] = sigma.m.2

# inverse:
VARi = solve(VAR)

# calculate lambdas:
lambda = VARi*sigma.e.2
la1 = lambda[1,1]; la2 = lambda[1,2]; la3 = lambda[1,3]
la4 = lambda[2,2]; la5 = lambda[2,3]
la6 = lambda[3,3]

### construction of the inverse of the gametic relationship matrix with the fortran
### program "GGRinv" (as 3-column lower triangle):
setwd("ProFor") # change to the ProFor directory containing "GGRinv"

# prepare the dataset needed for "GGRinv". The dataset must provide the following
# columns: "id", "dad", "mom", "indicator", "F"
data_in = cbind(data.ped[,1:3], rep(1, nrow(data.ped)), data.ped$F)
colnames(data_in)[c(4,5)] = c("indicator", "F")
data_in[is.na(data.ped$y) == FALSE, colnames(data_in) == "indicator"] = 2

data_in # note the indicator vector: 2s for gametic effects and 1s for transmitting ability

##      id dad mom indicator      F
## 1    1  0  0          1 0.000000
## 2    2  0  0          1 0.000000
## 3    3  1  0          1 0.000000
## 4    4  1  0          1 0.000000
## 5    5  3  4          1 0.125000
## 6    6  3  4          2 0.125000
## 7    7  1  4          2 0.250000
## 8    8  1  4          2 0.250000
## 9    9  3  2          1 0.000000
## 10   10  5  2          2 0.000000
## 11   11 10  9          2 0.203125
## 12   12  5  8          2 0.281250
## 13   13 10  9          2 0.203125
## 14   14  5  8          2 0.281250

# write dataset to ProFor directory:
write.table(data_in, "pedigree_input.txt", col.names=FALSE, row.names=FALSE, quote=FALSE, sep="\t")

# run fortran program to construct G with system function:

#####
#
# system("GGRinv > out.txt") #
#
#####

# move inverse of G (AImatrix.giv) to the ASReml directory so it will be available for the

```

```

# subsequent analysis with the ASReml software:

# system("mv AImatrix.giv ../ASreml")

### read in cross reference table (output file from the fortran program):
cross = read.table("crossref.txt",header = FALSE)
anzgl = max(max(cross$V4),max(cross$V5))
anzgl # this is going to be the number of equations for one random effect

## [1] 22

### rewrite inverse gametic relationship matrix to a nan matrix:
setwd("../ASreml") # change to the ASreml directory
GAMinv = read.table("AImatrix.giv",header=FALSE)
GAMi = matrix(ncol=anzgl, nrow=anzgl, 0)

i = 0
for(i in 1:nrow(GAMinv)){
  GAMi[ GAMinv[i,1] , GAMinv[i,2]] = GAMinv[i,3]
  GAMi[ GAMinv[i,2], GAMinv[i,1]] = GAMinv[i,3]
}
GAMi[1:10,1:10]

##           [,1] [,2]      [,3]      [,4]      [,5] [,6] [,7] [,8] [,9]
## [1,]  7.333333  0 -1.333333 -1.333333  0.00000  0  0  -2  0
## [2,]  0.000000  5  1.000000  0.000000  0.00000  0  0  0  0
## [3,] -1.333333  1  6.666667  1.000000 -2.00000 -2  0  0  0
## [4,] -1.333333  0  1.000000  9.666667 -2.00000  0 -2  0 -2
## [5,]  0.000000  0 -2.000000 -2.000000 10.85714  0  0  0  0
## [6,]  0.000000  0 -2.000000  0.000000  0.00000  2  0  0  0
## [7,]  0.000000  0  0.000000 -2.000000  0.00000  0  2  0  0
## [8,] -2.000000  0  0.000000  0.000000  0.00000  0  0  2  0
## [9,]  0.000000  0  0.000000 -2.000000  0.00000  0  0  0  2
## [10,] -2.000000  0  0.000000  0.000000  0.00000  0  0  0  0
##           [,10]
## [1,] -2.000000
## [2,]  0.000000
## [3,]  0.000000
## [4,]  0.000000
## [5,]  0.000000
## [6,]  0.000000
## [7,]  0.000000
## [8,]  0.000000
## [9,]  0.000000
## [10,]  3.333333

#
### construction of incidence matrices to set up the mixed model equations:
data.ped = cbind(data.ped, cross$V4, cross$V5)
colnames(data.ped) = c("id", "dad", "mom", "F", "y", "z1", "z2")

# construct Z.hs and Z.hd:
Z.hs = matrix(ncol = ncol(GAMi), nrow = length(which(is.na(data.ped$y) == FALSE)), 0)
Z.hd = matrix(ncol = ncol(GAMi), nrow = length(which(is.na(data.ped$y) == FALSE)), 0)
xc = 1; xr = 1

```

```

for(i in 1:nrow(data.ped)){
  if(data.ped[i,colnames(data.ped) == "z1"] ==
    data.ped[i,colnames(data.ped) == "z2"]) xc = xc + 1
  if(data.ped[i,colnames(data.ped) == "z1"] !=
    data.ped[i,colnames(data.ped) == "z2"]) {
    Z.hs[xr,xc] = 1; xc = xc + 1
    Z.hd[xr,xc] = 1; xc = xc + 1
    xr = xr + 1
  }
}
Z.hs # note that records of animals are linked to their expressed paternal gametes

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    1    0    0    0    0    0    0    0
## [2,]    0    0    0    0    0    0    0    1    0    0    0    0    0
## [3,]    0    0    0    0    0    0    0    0    0    1    0    0    0
## [4,]    0    0    0    0    0    0    0    0    0    0    0    0    1
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## [1,]      0      0      0      0      0      0      0      0      0
## [2,]      0      0      0      0      0      0      0      0      0
## [3,]      0      0      0      0      0      0      0      0      0
## [4,]      0      0      0      0      0      0      0      0      0
## [5,]      0      1      0      0      0      0      0      0      0
## [6,]      0      0      0      1      0      0      0      0      0
## [7,]      0      0      0      0      0      1      0      0      0
## [8,]      0      0      0      0      0      0      0      1      0

```

Z.hd # note that records of animals are linked to their expressed maternal gametes

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    0    0    0    1    0    0    0    0    0    0
## [2,]    0    0    0    0    0    0    0    0    1    0    0    0    0
## [3,]    0    0    0    0    0    0    0    0    0    0    1    0    0
## [4,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## [1,]      0      0      0      0      0      0      0      0      0
## [2,]      0      0      0      0      0      0      0      0      0
## [3,]      0      0      0      0      0      0      0      0      0
## [4,]      1      0      0      0      0      0      0      0      0
## [5,]      0      0      1      0      0      0      0      0      0
## [6,]      0      0      0      0      1      0      0      0      0
## [7,]      0      0      0      0      0      0      1      0      0
## [8,]      0      0      0      0      0      0      0      0      1

```

```

# construct Z.m:
# to construct Z.m the gametic effect numbers of the dams of the individuals with records
# are needed. These numbers are provided in column 8 and 9 of the cross-reference table

```

(crossref.txt) which was generated with GGRinv when the inverse of G was constructed:

```
colnames(cross)=c("ID","dad","mom","GoF","GoM","GfF","GfM","GmF","GmM")
crossY=cross[is.na(data.ped$y)==FALSE,]

Z.m = matrix(ncol = ncol(GAMi), nrow = length(which(is.na(data.ped$y) == FALSE)), 0)
for(i in 1:length(beob)){
  if(crossY[i,colnames(crossY) == "GmF"] ==
    crossY[i,colnames(crossY) == "GmM"]){ Z.m[i,crossY[i,colnames(crossY)=="GmF"]] = 2
  if(crossY[i,colnames(crossY) == "GmF"] !=
    crossY[i,colnames(crossY) == "GmM"]){
    Z.m[i,crossY[i,colnames(crossY)=="GmF"]] = 1
    Z.m[i,crossY[i,colnames(crossY)=="GmM"]] = 1
  }
}
```

*# note that the records of animals are linked to the transmitting ability of their dam (entry of 2
at this position) and to the two gametic effects of their dams if the dam itself has a phenotype
(entries of 1, respectively):*

Z.m

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]    0    0    0    2    0    0    0    0    0    0    0    0    0
## [2,]    0    0    0    2    0    0    0    0    0    0    0    0    0
## [3,]    0    0    0    2    0    0    0    0    0    0    0    0    0
## [4,]    0    2    0    0    0    0    0    0    0    0    0    0    0
## [5,]    0    0    0    0    0    0    0    0    0    0    0    2    0
## [6,]    0    0    0    0    0    0    0    0    0    1    1    0    0
## [7,]    0    0    0    0    0    0    0    0    0    0    0    2    0
## [8,]    0    0    0    0    0    0    0    0    0    1    1    0    0
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## [1,]      0      0      0      0      0      0      0      0      0
## [2,]      0      0      0      0      0      0      0      0      0
## [3,]      0      0      0      0      0      0      0      0      0
## [4,]      0      0      0      0      0      0      0      0      0
## [5,]      0      0      0      0      0      0      0      0      0
## [6,]      0      0      0      0      0      0      0      0      0
## [7,]      0      0      0      0      0      0      0      0      0
## [8,]      0      0      0      0      0      0      0      0      0
```

X

```
X = matrix(ncol = 1, nrow = length(beob),1);X
```

```
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
## [4,]    1
## [5,]    1
## [6,]    1
## [7,]    1
## [8,]    1
```

y

```
y = as.matrix(data.ped[is.na(data.ped$y) == FALSE, colnames(data.ped) == "y"])
```

y

```
##      [,1]
## [1,]  444
## [2,]  555
## [3,]  550
## [4,]  580
## [5,]  625
## [6,]  375
## [7,]  400
## [8,]  355
```

set up the mixed model equations:

```
LHS = rbind(cbind((t(X)%*%X), # left hand side
                  (t(X)%*%Z.hs),
                  (t(X)%*%Z.hd),
                  (t(X)%*%Z.m)),
            cbind((t(Z.hs)%*%X),
                  (t(Z.hs)%*%Z.hs+GAMi*la1),
                  (t(Z.hs)%*%Z.hd+GAMi*la2),
                  (t(Z.hs)%*%Z.m+GAMi*la3)),
            cbind((t(Z.hd)%*%X),
                  (t(Z.hd)%*%Z.hs+GAMi*la2),
                  (t(Z.hd)%*%Z.hd+GAMi*la4),
                  (t(Z.hd)%*%Z.m+GAMi*la5)),
            cbind((t(Z.m)%*%X),
                  (t(Z.m)%*%Z.hs+GAMi*la3),
                  (t(Z.m)%*%Z.hd+GAMi*la5),
                  (t(Z.m)%*%Z.m+GAMi*la6)))
```

```
RHS = rbind((t(X)%*%y),t(Z.hs)%*%y,t(Z.hd)%*%y,t(Z.m)%*%y) # right hand side
```

Solving the mixed model equations by direct inversion of the coefficient matrix

-> solutions in LS

```
LS = solve(LHS)%*%RHS
```

```
#####
##### ASREML: #####
#####
```

*# Now, the results generated in R (saved in vector LS) should be replicated with ASreml.
For this purpose, the dataset needed for ASreml can be prepared in R.*

*# The dataset must contain the columns ID = individual; g1nr = paternal gamete; g2nr =
maternal gamete; mor1 = paternal gamete of mother; mor2 = maternal gamete of mother;
and the phenotype:*

```
PP = as.data.frame(cbind(cross$ID, cross$GoF, cross$GoM, cross$GmF, cross$GmM,data.ped$y))
colnames(PP) = c("ID","g1nr","g2nr","mor1","mor2","phenotype")
PP[is.na(PP$phenotype) == TRUE, colnames(PP) == "phenotype"] = 0
```

```
write.table(PP,"dataset.txt",col.names=TRUE, row.names=FALSE, sep="\t", quote=FALSE)
```

```
#####
#
# system("asreml -l command_file.as") #
#
#####

#####

### Now, compare results of R and ASreml:
ASR = read.table("command_file.sln",header=FALSE) # <-- read in ASReML-results

results=as.data.frame(cbind(round(ASR[,3],3),round(LS,3)))
colnames(results)=c("ASreml_solution","R_solution")
results
```

##	ASreml_solution	R_solution
## 1	502.900	502.916
## 2	-4.730	-4.730
## 3	9.172	9.172
## 4	-8.276	-8.276
## 5	-3.118	-3.118
## 6	-10.370	-10.367
## 7	-11.700	-11.699
## 8	-6.700	-6.700
## 9	0.731	0.731
## 10	2.596	2.596
## 11	-10.480	-10.480
## 12	-8.535	-8.535
## 13	0.662	0.662
## 14	-5.966	-5.966
## 15	14.430	14.429
## 16	13.650	13.652
## 17	10.520	10.518
## 18	-15.910	-15.905
## 19	-14.470	-14.474
## 20	-4.952	-4.952
## 21	-8.947	-8.947
## 22	-17.400	-17.401
## 23	-15.820	-15.816
## 24	-4.417	-4.417
## 25	9.158	9.158
## 26	-8.400	-8.400
## 27	-3.129	-3.129
## 28	-10.650	-10.650
## 29	-11.980	-11.982
## 30	-6.885	-6.885
## 31	1.297	1.297
## 32	2.863	2.863
## 33	-9.806	-9.806
## 34	-8.152	-8.152
## 35	0.592	0.592
## 36	-6.046	-6.046
## 37	14.670	14.670
## 38	14.170	14.168

```
## 39      10.930      10.928
## 40     -16.440     -16.445
## 41     -14.190     -14.187
## 42      -5.296      -5.296
## 43      -9.484      -9.484
## 44     -18.010     -18.010
## 45     -15.590     -15.595
## 46      -9.293      -9.293
## 47       9.642       9.642
## 48      -6.765      -6.765
## 49      -3.052      -3.052
## 50      -6.669      -6.669
## 51      -8.056      -8.056
## 52      -4.248      -4.248
## 53      -7.235      -7.235
## 54      -1.143      -1.143
## 55     -20.300     -20.296
## 56     -14.250     -14.251
## 57       1.665       1.665
## 58      -5.010      -5.010
## 59      11.400      11.404
## 60       6.748       6.748
## 61       4.957       4.957
## 62      -8.757      -8.757
## 63     -18.930     -18.933
## 64      -0.265      -0.265
## 65      -1.545      -1.545
## 66      -9.321      -9.321
## 67     -19.380     -19.381
```

```
#####
# use solutions to calculate the breeding values, maternal effects and imprinting effects:
LSO = LSo[-1]
LSV = LSo[1:anzgl]
LSM = LSo[(anzgl+1):(anzgl*2)]
LSMatGen = LSo[(anzgl*2+1):length(LSo)]

genEff = cbind(LSV, LSM, LSMatGen)
genEff = as.data.frame(genEff)

genEff$eff.art = rep(NA,nrow(genEff)); Za = 1
for(i in 1:nrow(data.ped)){
  if(is.na(data.ped[i, colnames(data.ped) == "y"])) == TRUE){
    genEff[Za, colnames(genEff) == "eff.art"] = "TA"
    Za = Za + 1
  }
  if(is.na(data.ped[i, colnames(data.ped) == "y"])) == FALSE){
    genEff[Za, colnames(genEff) == "eff.art"] = "Gam"
    genEff[Za + 1, colnames(genEff) == "eff.art"] = "Gam"
    Za = Za + 2
  }
}

breedingvalues = as.data.frame(matrix(nrow = nrow(data.ped), ncol = 3, NA))
```

```

colnames(breedingvalues) = c("BVasV", "BVasM", "MatGenEff")

Za = 1
for(i in 1:nrow(data.ped)){
  if(genEff[Za, colnames(genEff) == "eff.art"] == "TA"){
    breedingvalues[i,1] = genEff[Za, colnames(genEff) == "LSV"]*2
    breedingvalues[i,2] = genEff[Za, colnames(genEff) == "LSM"]*2
    breedingvalues[i,3] = genEff[Za, colnames(genEff) == "LSMatGen"]*2
    Za = Za + 1
  } else{
    breedingvalues[i,1] = genEff[Za, colnames(genEff) == "LSV"] +
      genEff[Za+1, colnames(genEff) == "LSV"]
    breedingvalues[i,2] = genEff[Za, colnames(genEff) == "LSM"] +
      genEff[Za+1, colnames(genEff) == "LSM"]
    breedingvalues[i,3] = genEff[Za, colnames(genEff) == "LSMatGen"] +
      genEff[Za+1, colnames(genEff) == "LSMatGen"]
    Za = Za + 2
  }
}

EIE = breedingvalues$BVasM - breedingvalues$BVasV # imprinting effects

cbind(breedingvalues, EIE)

```

```

##      BVasV      BVasM  MatGenEff      EIE
## 1  -9.460519 -8.833437 -18.586248  0.62708237
## 2   18.344530  18.316064  19.284852 -0.02846514
## 3 -16.551165 -16.799683 -13.530692 -0.24851769
## 4   -6.235342  -6.257669  -6.103411 -0.02232742
## 5 -20.733189 -21.300473 -13.337829 -0.56728457
## 6 -18.399180 -18.867323 -12.304410 -0.46814369
## 7   3.326817   4.159899  -8.377387  0.83308169
## 8 -19.014511 -17.957677 -34.546803  1.05683454
## 9   1.323022   1.183603   3.329034 -0.13941945
## 10  8.463020   8.624277   6.393632  0.16125705
## 11 24.169643  25.096035  11.705235  0.92639223
## 12 -30.378976 -30.632113 -27.689358 -0.25313686
## 13 -13.898981 -14.780520  -1.810512 -0.88153934
## 14 -33.217307 -33.604973 -28.701756 -0.38766525

```

```

setwd("../")

```

Supplement 2

Example pedigree

Lower triangular matrix (decomposed inverse generalized gametic relationships)

Supplement 2

Example pedigree for a generalized relationship matrix, including all twelve cases. First three columns comprise pedigree information (animal identification, sire, dam; unknown animals are coded as zeros), the fourth column indicates the number of genetic effects for each individual (1: an average gametic effect; 2 two single gametic effects) and in the fifth column is the inbreeding coefficient. One or two (according to the number of genetic effects) abbreviations for cases (as explained in the manuscript) are given at the end of each row. There are three animals with two genetic effects and nine with a single one. While there are only twelve animals the resulting generalized gametic relationship matrix has dimension 15X15, one row and column for each genetic effect. The example in Figure S1.1 of a lower triangular matrix $(\mathbf{T}')^{-1}$ from a decomposed inverse of a generalized gametic relationship matrix was built from this pedigree.

1	0	0	1	0.00	a-00
2	0	0	2	0.00	g-00 g-00
3	1	2	1	0.00	a-agg
4	0	3	1	0.00	a-0a
5	1	0	1	0.00	a-a0
6	4	2	2	0.125	g-a g-gg
7	6	2	2	0.3125	g-gg g-gg
8	6	2	1	0.3125	a-gggg
9	1	3	1	0.25	a-aa
10	7	3	1	0.25	a-gga
11	6	0	1	0.00	a-gg0
12	0	2	1	0.00	a-0gg

Figure S2.1: Example of a lower triangular matrix $(\mathbf{T}')^{-1}$ from a decomposed inverse of a generalized gametic relationship matrix. Each row of the matrix pertains to a particular genetic effect. The last column indicates the respective combination of each kind of genetic effect (a: transmitting ability; g: gametic effect) with the genetic effects of the parents (a: transmitting ability; gg: pair of gametic effects; 0: unknown parent, and combinations thereof).

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a - 00
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g - 0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	g - 0
-0.5	-0.25	-0.25	1	0	0	0	0	0	0	0	0	0	0	0	0	a - agg
0	0	0	-0.5	1	0	0	0	0	0	0	0	0	0	0	0	a - 0a
-0.5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	a - a0
0	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	g - a
0	-0.5	-0.5	0	0	0	0	1	0	0	0	0	0	0	0	0	g - gg
0	0	0	0	0	0	-0.5	-0.5	1	0	0	0	0	0	0	0	g - gg
0	-0.5	-0.5	0	0	0	0	0	0	1	0	0	0	0	0	0	g - gg
0	-0.25	-0.25	0	0	0	-0.25	-0.25	0	0	1	0	0	0	0	0	a - gggg
-0.5	0	0	-0.5	0	0	0	0	0	0	0	1	0	0	0	0	a - aa
0	0	0	-0.5	0	0	0	0	-0.25	-0.25	0	0	1	0	0	0	a - gga
0	0	0	0	0	0	-0.25	-0.25	0	0	0	0	0	1	0	0	a - gg0
0	-0.25	-0.25	0	0	0	0	0	0	0	0	0	0	0	0	1	a - 0gg

Supplement 3

Detailed description of mouse data analyses and results

Material

Mouse body mass example analyses

The mouse line DUKs had been kept at the Leibniz-Institute for Farm Animal Biology without any intentional selection over 146 generations as a control for a long-term selection experiment on growth (Bünger *et al.*, 1998). Young DUKs mice were weaned at 21 days of age and from each litter up to 5 males and 4 females were then further reared in two separate cages. At an age of 42 days the body mass (BM42) of two randomly chosen males was measured (BM42 varied from 9.33g to 41.73g with an average of 29.38g). Next generations were formed by randomly choosing males and females from about 50% of all litters. This percentage varied from generation to generation due to random fluctuations in the pregnancy rate. As a general rule males with a record for BM42 were not used as sires, with some exceptions when there were too few other males left in a litter. So the vast majority (97.67%) of all 13,077 observations was from final progeny. The average BM42 did not change markedly over generations (Fig. 1). Pedigree size, including all animals with records and their ancestors, was 28,150 animals. Founders were assigned to generation zero. Ancestors were started to be recorded in the ninth generation. Inbreeding increased up to an average inbreeding coefficient of 0.62 (Fig. 2) in the last generation (generation number 153). The average inbreeding coefficient over all generations was 0.40. Data were analyzed by including 110 generations as fixed effects as well as 6,648 uncorrelated random litter effects. The genetic part of the model was alternatively covered by I) breeding values in an animal model (AM); II) by a classical gametic model (ICM); III) by a generalized gametic model (IGM); and IV) by a reduced version of the IGM (IRM). The IGM comprised gametic effects only for animals with records and transmitting abilities for all others. For the IRM all phenotyped animals without offspring (12,772 in number) were excluded from the underlying pedigree leaving only phenotyped parents (305 animals) and their ancestors (15,378 animals). While models ICM, IGM, and IRM consider genomic imprinting by design, the models were also applied under the assumption of non-imprinted Mendelian inheritance (MCM, MGM, and MRM). Inverse relationships of all kinds were computed using an own Fortran program and REML estimates of variance components were obtained with the help of the software package ASReml version 4.1 (Gilmore *et al.*, 2015). The R-packages “pedigree” version 1.4 (Coster, 2013) and “readxl” version 1.3.1 (Wickham *et al.*, 2019) in R version 4.0.0 were used to prepare the data.

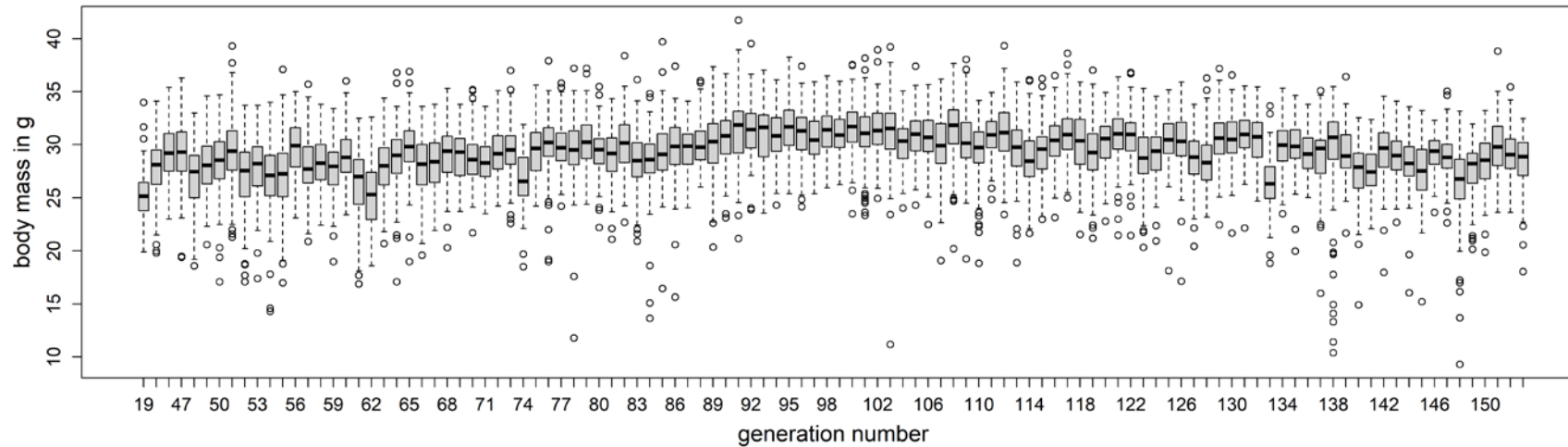


Figure S3.1. Distribution of body mass in g at an age of 42 days (BM42) in the DUKs mouse line over generations. Note, BM42 was measured in generation 19 and generation 23. Consecutive measurements are then available from generation 46 to generation 153.

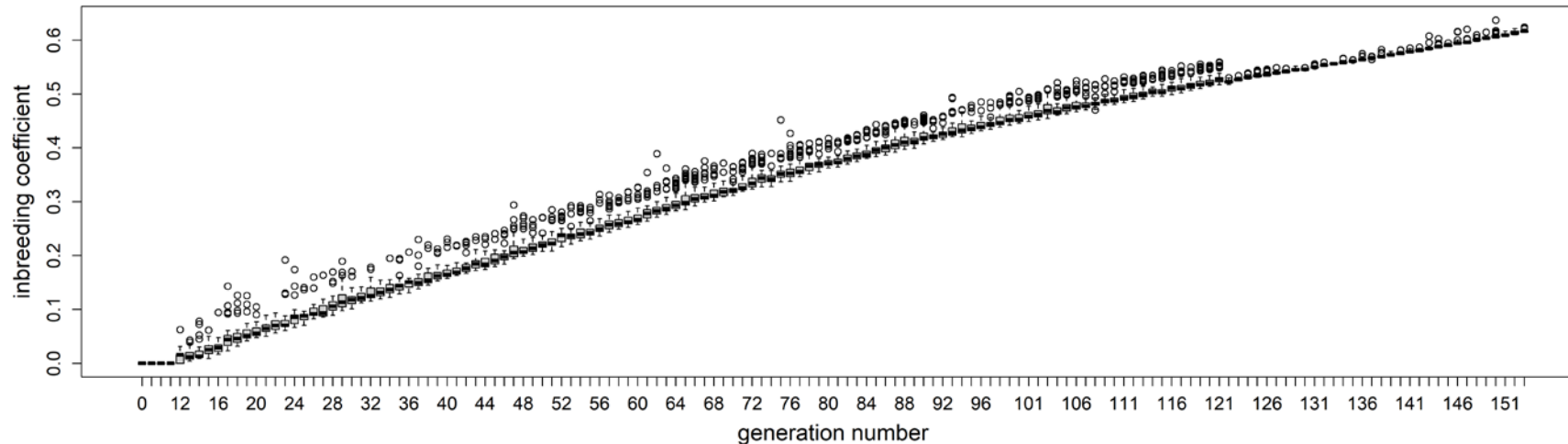


Figure S3.2. Distribution and development of the inbreeding coefficient in the DUKs mouse line over generations. Founders were assigned to a generation number of zero. Ancestors were started to be recorded in the ninth generation until a generation number of 153.

Methods

Mixed models

The AM used to analyze BM42 was:

$$y_{ijkl} = g_j + a_k + c_l + e_{ijkl} ,$$

where y_{ijkl} is the record of animal i ; g is the fixed effect of generation j ; a is the random effect of animal k ; c is the random effect of litter l ; and e_{ijkl} is the residual. In the ICM, the random effect of animal k is replaced by the effect of its paternal gamete g_s and by the effect of its maternal gamete g_d . The imprinting effect is defined as the vector of differences ($g_s - g_d$). The variance-covariance of random effects is assumed to be

$$\text{Var} \begin{bmatrix} g_s \\ g_d \\ e \end{bmatrix} = \begin{bmatrix} G\sigma_s^2 & G\sigma_{sd} & \mathbf{0} \\ G\sigma_{sd} & G\sigma_d^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I\sigma_e^2 \end{bmatrix} ,$$

where σ_e^2 is the residual variance and I is an identity matrix. Matrix G is the classical gametic relationship matrix (Gibson *et al.*, 1988; Schaeffer *et al.*, 1989). The components σ_s^2 and σ_d^2 are the gametic variances so that the corresponding variance of the imprinting effect is $\sigma_i^2 = \sigma_s^2 + \sigma_d^2 - \sigma_{sd}$ which represents the imprinting variance. Where no imprinting is observed $\sigma_s^2 = \sigma_d^2 = \sigma_{sd}$ and $\sigma_i^2 = 0$.

The IGM contains the gametic effect vectors g_s and g_d replaced by their transformed counterparts a_s and a_d , and, consequently, uses the corresponding relationship matrix $\bar{G}$ instead of G . The vectors a_s and a_d contain transmitting abilities (average gametic effects) for animals without records and gametic effects for all animals with records. The matrix $\bar{G}$ used for the IRM only contains the relationships between phenotyped parents and their ancestors. For the IRM a diagonal matrix W of weights is needed. Weights are $w_i = 1$ for observations from parents and

$$w_i = \left[\frac{\frac{1}{2}(1 - F_{si})\sigma_s^2 + \frac{1}{2}(1 - F_{di})\sigma_d^2 + \sigma_e^2}{\sigma_e^2} \right]^{-1}$$

for the final progeny. In the above equation F_s and F_d are the inbreeding coefficients of the sire and dam of an individual i . Eventually, random genetic effects and residuals of the IRM are assumed to have a variance-covariance of

$$Var \begin{bmatrix} \mathbf{a}_s \\ \mathbf{a}_d \\ \boldsymbol{\varepsilon} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{G}}\sigma_s^2 & \bar{\mathbf{G}}\sigma_{sd} & \mathbf{0} \\ \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_d^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}\sigma_e^2 \end{bmatrix}.$$

The significance of the imprinting variance was tested by comparing the logarithmic value of the restricted maximum likelihood (REML log-likelihood) of the imprinting models (ICM, IGM, IRM) to the REML log-likelihood outcomes of the corresponding Mendelian models (MCM, MGM, MRM) using a REML log-likelihood ratio test (RLRT). The test statistic was assumed to be asymptotically distributed as a mixture of two χ^2 -distributions with a number of numerator degrees of freedom (DF) of 1 and 2 (Neugebauer *et al.*, 2010a,b).

To take maternal genetic variances into account, the corresponding effect was added to all models except to the reduced model versions. As an example the variance-covariance of random effects of the IGM including a maternal gametic effect m is

$$Var \begin{bmatrix} \mathbf{a}_s \\ \mathbf{a}_d \\ \mathbf{m} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{G}}\sigma_s^2 & \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_{sm} & 0 \\ \bar{\mathbf{G}}\sigma_{sd} & \bar{\mathbf{G}}\sigma_d^2 & \bar{\mathbf{G}}\sigma_{dm} & 0 \\ \bar{\mathbf{G}}\sigma_{sm} & \bar{\mathbf{G}}\sigma_{dm} & \bar{\mathbf{G}}\sigma_m^2 & 0 \\ 0 & 0 & 0 & \mathbf{I}\sigma_e^2 \end{bmatrix}.$$

The existence of maternal genetic variance was statistically tested by comparing the REML log-likelihood of an AM without maternal genetic effect to the REML log-likelihood of the AM with a maternal genetic effect. For this purpose a RLRT with one DF was used. When the maternal genetic variance turned out to be significant, it was added to all other models except to the reduced model versions.

Calculation of population parameters

For the AM the direct heritability was calculated as $h^2 = \sigma_a^2 / \sigma_p^2$, where σ_a^2 is the additive genetic variance and σ_p^2 is the phenotypic variance with $\sigma_p^2 = \sigma_a^2 + \sigma_c^2 + \sigma_e^2$, where σ_c^2 is the litter variance. The relative litter variance was calculated as $c^2 = \sigma_c^2 / \sigma_p^2$. When a maternal genetic effect m_a was added to the model, σ_p^2 was defined as $\sigma_p^2 = \sigma_a^2 + \sigma_{m_a}^2 + 2\sigma_{am_a} + \sigma_c^2 + \sigma_e^2$. The relative maternal variance was calculated as $m^2 = \sigma_{m_a}^2 / \sigma_p^2$.

For the Mendelian models the direct heritabilities were calculated as $h^2 = \sigma_a^2 / \sigma_p^2$, where $\sigma_a^2 = 2\sigma_g^2$ and $\sigma_p^2 = 2\sigma_g^2 + \sigma_c^2 + \sigma_e^2$, where σ_g^2 is the gametic variance. The latter expression of σ_p^2 was used to calculate c^2 ($c^2 = \sigma_c^2 / \sigma_p^2$). When a maternal effect m was added, σ_p^2 was defined as $\sigma_p^2 = 2\sigma_g^2 + 2\sigma_m^2 + 4\sigma_{gm} + \sigma_c^2 + \sigma_e^2$. The relative maternal variance was then calculated as $m^2 = \sigma_{ma}^2 / \sigma_p^2$, where $\sigma_{ma}^2 = 2\sigma_m^2$.

For the imprinting models the direct heritabilities were calculated as $h^2 = \sigma_a^2 / \sigma_p^2$, where $\sigma_a^2 = \sigma_s^2 + \sigma_d^2$ and $\sigma_p^2 = \sigma_s^2 + \sigma_d^2 + \sigma_e^2 + \sigma_c^2$. This expression of σ_p^2 was used to calculate c^2 as $c^2 = \sigma_c^2 / \sigma_p^2$. When a maternal effect was added, σ_p^2 was defined as $\sigma_p^2 = \sigma_s^2 + \sigma_d^2 + 2\sigma_m^2 + 2\sigma_{sm} + 2\sigma_{dm} + \sigma_c^2 + \sigma_e^2$. The relative maternal variance was then calculated as $m^2 = \sigma_{ma}^2 / \sigma_p^2$, where $\sigma_{ma}^2 = 2\sigma_m^2$. The relative imprinting variance was calculated as $i^2 = \sigma_i^2 / \sigma_a^2 = \sigma_i^2 / (\sigma_s^2 + \sigma_d^2)$.

Technical notes

Half of the additive genetic variance estimated with the AM was used as starting value for the Mendelian models. The variance components estimated with the Mendelian models were used as starting values for the imprinting models, respectively.

The initial weight used for the MRM was calculated based on the variance estimates of the MCM. Furthermore, these variance estimates were used as starting values for the first run of the MRM. The same procedure was used for the IRM but based on the estimates of the ICM. The analyses with the reduced model versions (MRM, IRM) were repeated with adapted weightings until the REML log-likelihood outcomes stabilized. In addition, to show equivalence between models, the MRM and IRM were applied with an iteration number fixed to one.

Results

The REML log-likelihood outcomes as well as the genetic parameters were equal for the Mendelian inheritance models and for the imprinted inheritance models, respectively. The reduced model versions (MRM, IRM) led to the same REML log-likelihood as the gametic and generalized model versions when the number of iterations was fixed to one (Table S3.1). The size of the system of equations could be reduced for the imprinted inheritance models from 112,600 equations (ICM), over 82,454 equations (IGM), to 31,366 equations (IRM) (Table S3.1). In comparison, the system of equations of the AM comprised 28,150 equations. The total number of non-zero elements in the lower triangle of the inverted variance-covariance

matrix of the random genetic effects could be reduced from 184,046 non-zero elements (ICM) to 51,372 non-zero elements (IRM).

Excluding maternal effects, the imprinting analyses resulted in significant imprinting variances. Comparing the REML log-likelihood outcomes (Table S3.1) of the imprinting models with the REML log-likelihood outcomes of the corresponding Mendelian models led to an RLRT of 46.06 with $P = 9.96 \times 10^{-11}$. The comparison of the REML log-likelihood outcomes of the MRM and IRM resulted into a RLRT of 38.74 with $P = 3.87 \times 10^{-9}$. Using the ICM and IGM, POEs explained 31.40% ($\pm 9.40\%$) of the additive genetic variance while the IRM led to a ratio of 39.10% ($\pm 8.30\%$). The parental contributions of gametes to the imprinting variance can be calculated as $(\sigma_s^2 - \sigma_{sd})/\sigma_i^2$ for the paternal contribution and $(\sigma_d^2 - \sigma_{sd})/\sigma_i^2$ for the maternal contribution. The covariance between paternal and maternal effects was larger than the paternal variance thereby resulting in negative paternal contributions of -36.91% ($\pm 28.88\%$) for the ICM and IGM. Moreover, a negative paternal contribution was found for the IRM with -16.57% ($\pm 16.68\%$).

Heritability was estimated to be almost 50.00% ($\pm 3.00\%$) for the non-imprinted Mendelian inheritance models and reached almost 57.00% ($\pm 2.50\%$) for the reduced version. Assuming imprinted inheritance, heritability estimates ranged from 55.80% ($\pm 3.30\%$) for the ICM and IGM to 66.60% ($\pm 2.50\%$) for the IRM. All estimated genetic parameters can be found in Table S3.1. Estimates of variance and covariance components can be found in Table S3.2. Figure S3.3 shows only little genetic progress of about 5g in 146 generations for BM42.

Using the AM, the maternal genetic variance was significant with a RLRT of 69.18 with $P = 8.99 \times 10^{-17}$. With the non-imprinted Mendelian inheritance models (AM, MCM, MGM), h^2 decreased to 27.50% ($\pm 4.70\%$) when a maternal effect was included. The relative litter variance only slightly decreased from 27.50% ($\pm 1.50\%$) to 20.60% ($\pm 1.60\%$) when a maternal effect was added. The imprinting models that included a maternal effect (ICM, IGM) did not converge. Eventually, as the log-likelihood only decreased at the second decimal place at every 100 iterations, the program was terminated and the genetic parameters calculated (Table S3.1 and S3.2). In order to obtain convergence, estimates obtained from the ICM were used as starting values for ICM and IGM implemented to Echidna v1.32, a mixed model software package that is similar to ASReml also using restricted maximum likelihood (Gilmour, 2018). Presumably based on another convergence criterion, Echidna converged at a REML log-likelihood of $-17,855.95$ for both models. At a 5% significance level, the imprinting variance is

thereby not significant when a maternal effect is included (RLRT = 2.64; DF = 3; $P = 0.45$). In contrast, the maternal genetic variance turned out to be significant (RLRT = 25.76; DF = 3; $P = 1.07 \times 10^{-5}$).

While heritability and relative litter variance further decreased, an increase in relative maternal variance could be observed (Table S3.1). With the inclusion of a maternal effect, the relative imprinting variance decreased from 31.40% to 25.60%. In contrast, the correlation between the two parental gametic effects increased from 0.82 to 0.91. While without maternal effect negative paternal contributions were observed for the ICM and IGM (−36.91%), the paternal contribution to the imprinting variance now exceeds 100% with 161.78%. The maternal contribution is thereby negative with −61.78%.

Table S3.1. The logarithmic values of the restricted maximum likelihood (*LogL*), overall number of random genetic effects (no. equa.), and the total number of non-zero elements (non zero) in the lower triangle of the inverted variance-covariance matrix of random genetic effects. Heritability (h^2), relative litter variance (c^2), relative imprinting variance (i^2), and correlation between gametic parental effects (r) are provided with their standard errors in brackets. The *LogL*, relative maternal genetic variance (m^2), h^2 and c^2 were additionally estimated with a model that included a maternal genetic effect (mat. effect)

	non-imprinted Mendelian inheritance				imprinted inheritance		
	AM	MCM	MGM	MRM	ICM	IGM	IRM
<i>LogL</i>	-17,891.870	-17,891.870	-17,891.870	-17,891.870 ^a -17,894.640 ^b	-17,868.840	-17,868.840	-17,868.840 ^a -17,875.270 ^b
no. equa.	28,150	56,300	41,227	15,683	112,600	82,454	31,366
non zero	92,023	184,046	103,125	51,372	184,046	103,125	51,372
h^2	0.497 (±0.030)	0.498 (±0.030)	0.498 (±0.030)	0.565 (±0.025) ^b	0.558 (±0.033)	0.558 (±0.033)	0.666 (±0.025) ^b
c^2	0.275 (±0.015)	0.275 (±0.015)	0.275 (±0.015)	0.248 (±0.019) ^b	0.242 (±0.017)	0.242 (±0.017)	0.196 (±0.020) ^b
i^2	-	-	-	-	0.314 (±0.094)	0.314 (±0.094)	0.391 (±0.083) ^b
r	-	-	-	-	0.819 (±0.123)	0.819 (±0.123)	0.713 (±0.102) ^b
mat. effect	AM	MCM	MGM	MRM	ICM	IGM	IRM
<i>LogL</i>	-17,857.280	-17,857.280	-17,857.280	-	-17,855.960	-17,855.960	-
m^2	0.127 (±0.034)	0.127 (±0.034)	0.127 (±0.034)	-	0.133 (±0.002)	0.133 (±0.002)	-
h^2	0.275 (±0.047)	0.275 (±0.047)	0.275 (±0.047)	-	0.196 (±0.002)	0.196 (±0.002)	-
c^2	0.206 (±0.016)	0.206 (±0.016)	0.206 (±0.016)	-	0.188 (±0.009)	0.188 (±0.010)	-
i^2	-	-	-	-	0.256 (not es.)	0.256 (not es.)	-
r	-	-	-	-	0.907 (not es.)	0.907 (not es.)	-

AM = animal model; MCM = classical gametic model that assumes non-imprinted Mendelian inheritance; MGM = generalized gametic model that assumes non-imprinted Mendelian inheritance; MRM = generalized reduced gametic model that assumes non-imprinted Mendelian inheritance; ICM = imprinting model with classical gametic relationships; IGM = imprinting model with generalized gametic relationships; IRM = imprinting model with generalized reduced gametic relationships.

^a number of iterations fixed to one; weightings were calculated based on variance component estimates from the corresponding classical models (MCM and ICM)

^b multiple runs with adapted weightings until *LogL* stabilized; multiple iterations per run were allowed in order to obtain convergence

not es. = standard error was not estimable because US structure was not positive definite. As a consequence, the variance of variance components was not available for this structure

Table S3.2. Variance-covariance-components for the mixed models that assume non-imprinted Mendelian inheritance or imprinted inheritance. Variance-covariance-components are also provided for models including a maternal genetic effect (*mat. effect*). Standard errors are given in brackets

	non-imprinted Mendelian inheritance				imprinted inheritance		
	AM	MCM	MGM	MRM ^a	ICM	IGM	IRM ^a
σ_a^2	4.284 (±0.355)	4.284 (±0.355)	4.284 (±0.355)	5.126 (±0.433)	4.953 (±0.408)	4.953 (±0.408)	6.429 (±0.517)
σ_p^2	8.607 (±0.218)	8.607 (±0.219)	8.607 (±0.219)	9.075 (±0.377)	8.871 (±0.236)	8.871 (±0.236)	9.660 (±0.434)
σ_e^2	1.956 (±0.112)	1.956 (±0.112)	1.956 (±0.112)	1.701 (±0.030)	1.774 (±0.125)	1.774 (±0.125)	1.335 (±0.024)
σ_c^2	2.367 (±0.105)	2.366 (±0.105)	2.366 (±0.105)	2.249 (±0.108)	2.144 (±0.119)	2.144 (±0.119)	1.896 (±0.127)
σ_g^2	-	2.142 (±0.177)	2.142 (±0.177)	2.563 (±0.216)	-	-	-
σ_i^2	-	-	-	-	1.555 (±0.519)	1.555 (±0.519)	2.514 (±0.618)
σ_s^2	-	-	-	-	1.125 (±0.247)	1.125 (±0.247)	1.541 (±0.288)
σ_{sd}	-	-	-	-	1.699 (±0.233)	1.699 (±0.233)	1.957 (±0.270)
σ_d^2	-	-	-	-	3.828 (±0.360)	3.828 (±0.360)	4.888 (±0.443)
mat. effect	AM	MCM	MGM	MRM ^a	ICM	IGM	IRM ^a
σ_a^2	2.567 (±0.432)	2.567 (±0.432)	2.567 (±0.432)	-	1.945 (±0.024)	1.945 (±0.034)	-
σ_p^2	9.341 (±0.356)	9.341 (±0.356)	9.341 (±0.356)	-	9.906 (±0.123)	9.906 (±0.123)	-
σ_e^2	2.429 (±0.134)	2.429 (±0.134)	2.429 (±0.134)	-	2.605 (±0.032)	2.605 (±0.045)	-
σ_c^2	1.925 (±0.126)	1.925 (±0.126)	1.925 (±0.126)	-	1.866 (±0.023)	1.866 (±0.093)	-
$\sigma_{m_a}^2$	1.189 (±0.302)	1.189 (±0.302)	1.189 (±0.302)	-	1.320 (±0.016)	1.320 (±0.023)	-
σ_{am_a}	0.615 (±0.295)	0.615 (±0.295)	0.615 (±0.295)	-	-	-	-
σ_g^2	-	1.284 (±0.216)	1.284 (±0.216)	-	-	-	-
σ_i^2	-	-	-	-	0.497 (±0.006)	0.497 (±0.009)	-
σ_s^2	-	-	-	-	1.528 (±0.019)	1.528 (±0.026)	-
σ_{sd}	-	-	-	-	0.724 (±0.009)	0.724 (±0.013)	-
σ_d^2	-	-	-	-	0.417 (±0.005)	0.417 (±0.007)	-
σ_{sm}	-	-	-	-	0.619 (±0.008)	0.619 (±0.011)	-
σ_{dm}	-	-	-	-	0.467 (±0.006)	0.467 (±0.008)	-

AM = animal model; MCM = classical gametic model that assumes non-imprinted Mendelian inheritance; MGM = generalized gametic model that assumes non-imprinted Mendelian inheritance; MRM = generalized reduced gametic model that assumes non-imprinted Mendelian inheritance; ICM = imprinting model with classical gametic relationships; IGM = imprinting model with generalized gametic relationships; IRM = imprinting model with generalized reduced gametic relationships

^a multiple runs with adapted weightings until restricted maximum likelihood stabilized; multiple iterations per run were allowed in order to obtain convergence

σ_a^2 = additive genetic variance; σ_p^2 = phenotypic variance; σ_e^2 = residual variance; σ_c^2 = litter variance; $\sigma_{m_a}^2$ = maternal genetic effect; σ_{am_a} = covariance between additive genetic variance and maternal genetic variance; σ_g^2 = gametic variance; σ_i^2 = imprinting variance; σ_s^2 = gametic variance as sire; σ_d^2 = gametic variance as dam; σ_{sd} = covariance between parental gametic variances; σ_{sm} = covariance between gametic variance as sire and maternal gametic variance; σ_{dm} = covariance between gametic variance as dam and maternal gametic variance

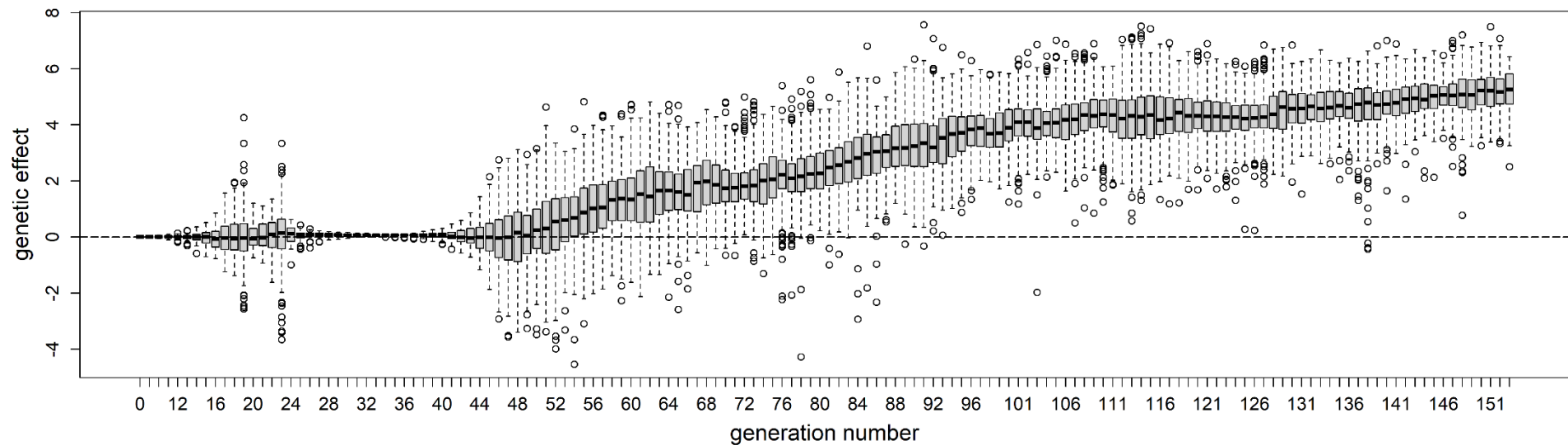


Figure S3.3. Development of breeding values shows the genetic trend of body mass at an age of 42 days in g (BM42) in the DUKs mouse line. Note, BM42 was measured in generation 19 and generation 23. Consecutive measurements were then available from generation 46 to generation 153.

References

- Bünger, L., U. Renne, G. Dietl, and S. Kuhla, 1998: Long-term selection for protein amount over 70 generations in mice. *Genet. Res.* **72**, 93-109.
- Coster, A., 2013: Pedigree: pedigree functions. R package version 1.4.
- Gibson, J. P., B. W. Kennedy, L. R. Schaeffer, and O. I. Southwood, 1988: Gametic models for estimation of autosomally inherited genetic effects that are expressed only when received from either a male or female parent. *J. Dairy Sci.* **71**, 143.
- Gilmour, A. R., B. J. Gogel, B. R. Cullis, S. J. Welham, and R. Thompson, 2015: ASReml User Guide Release 4.1 Functional Specification (VSN International Ltd, Hemel Hempstead).
- Gilmour, A. R., 2018: Echidna mixed models software. *Proceedings of the World Congress Applied to Livestock Genetics*, February 2018, Auckland, New Zealand, pp. 995.
URL <https://www.echidnamms.org/> (version 1.32; last accessed in September 2020).
- Neugebauer, N., H. Luther, and N. Reinsch, 2010: Parent-of-origin effects cause genetic variation in pig performance traits. *Animal* **4**, 672-681.
- Neugebauer, N., I. Rader, H. J. Schild, D. Zimmer, and N. Reinsch, 2010: Evidence for parent-of-origin effects on genetic variability of beef traits. *J. Anim. Sci.* **88**, 523-532.
- R Core Team, 2020: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/> (last accessed in September 2020).
- Schaeffer, L. R., B. W. Kennedy, and J. P. Gibson, 1989: The inverse of the gametic relationship matrix. *J. Dairy Sci.* **72**, 1266-1272.
- Wickham, H. et al., 2019: Package „readxl“ – Read Excel Files. R package version 1.3.1.
<https://cran.r-project.org/web/packages/readxl/readxl.pdf>
<https://readxl.tidyverse.org>
<https://github.com/tidyverse/readxl>