

ResponseToSelection__CarcassTraits__Multivariate

Silvan

18 10 2018

Goal

To compute genetic gains of the traits carcass fatt, carcass conformation and carcass weight. Each trait for adults and calves. (6 traits in total).

Plan

- Firstly, to record and name all input parameters either as tables or as variables in separate files.
- Secondly, show all necessary equations/descriptions from bottom to top.
- Thirdly, create R chunks that read the input parameters and solve the equations.

Material

- Residual, phenotypic and genetic variances and covariances of and between all traits.
- Breeding scenario (number of records of offspring per selection candidate). The scenario will result in a relationship matrix. For male and female selection paths each a separate relationship matrix.
- Generation intervall and proportion selected of the population for male and female selection paths.
- Economic value for each trait.

Implementation

First I import the input parameters using `read.table()`. Then I compute parameters that do not require equations. Then I compute all necessary steps using equations (derived from @Phocas1998).

Input parameters

Genetic, Residual and Phenotypic Variances/Covariances

With CC,CF,CW.

```
# import variance covariance dataframes and compute matrix
## genetic
genetic_var_cov <- "adults_calves_genetic_variances_covariances"
genetic_var_cov <- read.table(file = genetic_var_cov, sep=";", header=TRUE)
genetic_var_cov <- as.matrix(genetic_var_cov)
## residual
residual_var_cov <- "adults_calves_residual_variances_covariances"
residual_var_cov <- read.table(file = residual_var_cov, sep=";", header=TRUE)
```

```

residual_var_cov <- as.matrix(residual_var_cov)
## phenotypic
phenotypic_var_cov <- "adults_calves_phenotypic_variances_covariances"
phenotypic_var_cov <- read.table(file = phenotypic_var_cov, sep=";", header=TRUE)
phenotypic_var_cov <- as.matrix(phenotypic_var_cov)

```

Number of Offspring

First the number of relatives is imported.

```

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

## male
male_offspring <- "LM_number_of_offspring"
male_offspring <- read.table(file = male_offspring, sep=";", header=TRUE) %>%
  select(male)

## Warning in read.table(file = male_offspring, sep = ";", header = TRUE):
## incomplete final line found by readTableHeader on 'LM_number_of_offspring'
male_offspring <- as.numeric(male_offspring)
male_offspring

## [1] 9

## female
female_offspring <- "LM_number_of_offspring"
female_offspring <- read.table(file = female_offspring, sep=";", header=TRUE) %>%
  select(female)

## Warning in read.table(file = female_offspring, sep = ";", header = TRUE):
## incomplete final line found by readTableHeader on 'LM_number_of_offspring'
female_offspring <- as.numeric(female_offspring)
female_offspring

## [1] 2

```

Generation intervall L

```

library(dplyr)
## male
male_generationintervall <- "LM_generation_intervall"
male_generationintervall <- read.table(file = male_generationintervall, sep=";", header=TRUE) %>%
  select(male)

```

```
## Warning in read.table(file = male_generationintervall, sep = ";",
## header = TRUE): incomplete final line found by readTableHeader on
## 'LM_generation_intervall'

male_generationintervall <- as.numeric(male_generationintervall)
male_generationintervall

## [1] 6.2

## female
female_generationintervall <- "LM_generation_intervall"
female_generationintervall <- read.table(file = female_generationintervall, sep=";", header=TRUE) %>%
  select(female)

## Warning in read.table(file = female_generationintervall, sep = ";",
## header = TRUE): incomplete final line found by readTableHeader on
## 'LM_generation_intervall'

female_generationintervall <- as.numeric(female_generationintervall)
female_generationintervall

## [1] 5.2
```

Proportion selected

```
library(dplyr)
## male
male_proportionsselected <- "LM_proportion_selected"
male_proportionsselected <- read.table(file = male_proportionsselected, sep=";", header=TRUE) %>%
  select(male)

## Warning in read.table(file = male_proportionsselected, sep = ";",
## header = TRUE): incomplete final line found by readTableHeader on
## 'LM_proportion_selected'

male_proportionsselected <- as.numeric(male_proportionsselected)
male_proportionsselected

## [1] 0.6195652

## female
female_proportionsselected <- "LM_proportion_selected"
female_proportionsselected <- read.table(file = female_proportionsselected, sep=";", header=TRUE) %>%
  select(female)

## Warning in read.table(file = female_proportionsselected, sep = ";",
## header = TRUE): incomplete final line found by readTableHeader on
## 'LM_proportion_selected'

female_proportionsselected <- as.numeric(female_proportionsselected)
female_proportionsselected

## [1] 0.7058824
```

Economic values a

```
library(dplyr)
# always calves first and CC,CF,CW
a <- "LM_economic_values"
a <- read.table(file = a, sep=";", header=TRUE)

## Warning in read.table(file = a, sep = ";", header = TRUE): incomplete final
## line found by readTableHeader on 'LM_economic_values'

aMat <- as.matrix(a)
a <- as.vector(aMat)
dim(t(a))

## [1] 1 6

is.vector(a)

## [1] TRUE

dim(aMat)

## [1] 1 6
```

Computations without equations

Number of traits n

```
n <- as.numeric(nrow(phenotypic_var_cov))
```

Variance Covariance Matrices G (genetic) and R_y (residual)

```
G <- genetic_var_cov
male_R <- residual_var_cov %x% diag(male_offspring)
female_R <- residual_var_cov %x% diag(female_offspring)
```

Relationship Matrix A_y and Inverse A_y^{-1}

```
## male
# Compute the relationship matrix dependent on the number of offspring of the selection candidate. The
suppressPackageStartupMessages(library(pedigreeemm))

## Warning: package 'lme4' was built under R version 3.4.4
mnumb <- male_offspring+1
male_A <- pedigree(sire = c(NA,rep(1,times = male_offspring)), dam =c(NA,rep(NA,times = male_offspring))
# Compute inverse of relationship matrix.
male_Ainv <- getAInv(male_A)
## female
# Compute the relationship matrix dependent on the number of offspring of the selection candidate. The
suppressPackageStartupMessages(library(pedigreeemm))
mnumb <- female_offspring+1
female_A <- pedigree(sire = c(NA,rep(1,times = female_offspring)), dam =c(NA,rep(NA,times = female_offspring))
```

```
# Compute inverse of relationship matrix.
female_Ainv <- getAInv(female_A)
```

Design matrix Z_y

```
# male
male_Z<-cbind(matrix(0, nrow=male_offspring*n,ncol=n),diag(male_offspring*n))
dim(male_Z)

## [1] 54 60

# female
female_Z<-cbind(matrix(0, nrow=female_offspring*n,ncol=n),diag(female_offspring*n))
dim(female_Z)

## [1] 12 18
male_offspring*n

## [1] 54
male_offspring

## [1] 9
```

Selection intensity i_y and value of truncation x_y

```
# male
male_i <- dnorm(qnorm(1-male_proportionselected))/male_proportionselected
male_x <- qnorm(male_proportionselected, lower.tail = FALSE)
# female
female_i <- dnorm(qnorm(1-female_proportionselected))/female_proportionselected
female_x <- qnorm(female_proportionselected, lower.tail = FALSE)
```

Computations with Equations

Factor of variance reduction k_y

$$k_y = i_y(i_y - x_y)$$

```
# male
male_k <- male_i*(male_i-male_x)
# female
female_k <- female_i*(female_i-female_x)
```

Prediction error variance PEV_y

$$PEV_y = (Z_y^T R_y^{-1} Z_y + A_y^{-1} \otimes G^{-1})^{-1} [1 : n, 1 : n]$$

```
dim(male_Z)
```

```
## [1] 54 60
```

```

dim(male_Ainv)

## [1] 10 10
dim(male_R)

## [1] 54 54
# male
male_PEV <- solve(t(male_Z)%*%solve(male_R)%*%male_Z+male_Ainv%x%solve(G))[1:n,1:n]
# female
female_PEV <- solve(t(female_Z)%*%solve(female_R)%*%female_Z+female_Ainv%x%solve(G))[1:n,1:n]

```

Covariance matrix between true and estimated breeding values C_y

$$C_y = G - PEV_y$$

```

# male
male_C <- G-male_PEV
# female
female_C <- G-female_PEV

```

Asymptotic covariance matrix C_y^* between true and estimated breeding values

$$\begin{bmatrix} C_1^* \\ C_2^* \end{bmatrix} = \left(\begin{bmatrix} 1 + 1/2k_1 & 1/2k_2 \\ 1/2k_1 & 1 + 1/2k_2 \end{bmatrix}^{-1} \otimes I_n \right) \bullet \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

```

Ca <- (solve(matrix(c(1+0.5*male_k, 0.5*female_k,0.5*male_k, 1+0.5*female_k),nrow = 2, ncol = 2, byrow=
# male
male_Ca <- Ca[1:n,]
# female
female_Ca <- Ca[(1+n):(2*n),]

```

```
male_Ca
```

```

## 6 x 6 Matrix of class "dgeMatrix"
##           cCC           aCC           cCF           aCF           cCW
## [1,]  0.166238488  1.492515e-01 -6.449257e-03 -0.01319234  0.89538205
## [2,]  0.149251515  1.649402e-01  5.254334e-05  0.01282406  0.61108523
## [3,] -0.006449257  5.254334e-05  3.869418e-02  0.03530919 -0.02191305
## [4,] -0.013192337  1.282406e-02  3.530919e-02  0.04655004 -0.20109805
## [5,]  0.895382051  6.110852e-01 -2.191305e-02 -0.20109805  15.57528391
## [6,]  1.293278281  1.748051e+00  2.361637e-01  0.70702594  0.16158249
##           aCW
## [1,]  1.2932783
## [2,]  1.7480508
## [3,]  0.2361637
## [4,]  0.7070259
## [5,]  0.1615825
## [6,] 101.6735598

```

```
aMat
```

```

##           cCC           aCC           cCF           aCF           cCW           aCW
## [1,]  0.3860558  0.1961701  0.5761602  0.13559 -0.002289594 -0.002251035

```

Asymptotic genetic variance/covariance matrix G^*

$$G^* = G - 1/2 \sum_{y=1}^2 (k_y (a^T C_y^* a)^{-1} C_y^* a a^T C_y^*)$$

```
Ga<-G-0.5*(as.numeric(solve(as.numeric(male_k*t(a)%%male_Ca%%a)))*male_Ca%%a%%t(a)%%male_Ca+as.num
```

Accuracy of the estimated overall breeding value r_y^*

$$r_y^* = \sqrt{\frac{a^T C_y^* a}{a^T G^* a}}$$

```
# male
male_ra <- sqrt(as.numeric(t(a)%%male_Ca%%a)/as.numeric(t(a)%%Ga%%a))
# female
female_ra <- sqrt(as.numeric(t(a)%%female_Ca%%a)/as.numeric(t(a)%%Ga%%a))
male_ra
```

```
## [1] 1.112569
```

```
female_ra
```

```
## [1] 0.6054857
```

Standard deviation of the estimated overall breeding value σ^*

$$\sigma^* = \sqrt{a^T G^* a}$$

```
stdev_overall<-sqrt(as.numeric(t(a)%%Ga%%a))
stdev_overall
```

```
## [1] 0.2259687
```

Asymptotic genetic gain of the estimated overall breeding value

$$Q = \frac{\sum_{y=1}^2 i_y r_y^* \sigma^*}{\sum_{y=1}^2 L_y}$$

```
Qoverall_index <- (male_i*male_ra*stdev_overall+female_i*female_ra*stdev_overall)/(male_generationinter
Qoverall_index
```

```
## [1] 0.01941592
```

Correlated asymptotic genetic gain of traits when selecting for the aggregate genotype.

$$Q_m^* = \frac{\sum_{y=1}^2 i_y a^T (C_y^*)_m \sqrt{a^T C_y^* a}}{\sum_{y=1}^2 L_y}$$

```

QaCF<- (male_i*as.numeric(t(a)%*%male_Ca[, "aCF"])*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*as.nume
jQcCF <- (male_i*as.numeric(t(a)%*%male_Ca[, "cCF"])*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*as.nu
QaCC <- (male_i*as.numeric(t(a)%*%male_Ca[, "aCC"])*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*as.nu
QcCC <- (male_i*as.numeric(t(a)%*%male_Ca[, "cCC"])*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*as.nu
QaCW <- (male_i*as.numeric(t(a)%*%male_Ca[, "aCW"])*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*as.nu
QcCW <- (male_i*as.numeric(t(a)%*%male_Ca[, "cCW"])*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*as.nu

```

Correlated asymptotic genetic gain of a trait in the aggregate breeding value involving not all traits

$$b = \sigma_n^{-1}(G_n)^T a$$

Where n indicates the trait. Here aCF.

```

# B <- (1/G[which( colnames(G)=="aCF" ), "aCF"])*t(G[, "aCF"])*a
# B
# t(G[, "aCF"])*a
# aMat
# Ga[which( colnames(G)=="cCC" ), "aCF"]
# a[which( colnames(G)=="cCC" ), "aCF"]

```

$$Q_m^* = \frac{\sum_{y=1}^2 i_y b^T (C_y^*)_m \sqrt{b^T C_y^* b}}{\sum_{y=1}^2 L_y}$$

```

# aCFQ <- male_i*B[which(colnames(G)=="aCF")]*male_Ca[which( colnames(G)=="aCF" ), "aCF"]*sqrt(as.numeri
#
# aCFQ
#
# cCFQ <- male_i*B[which(colnames(G)=="cCF")]*male_Ca[which( colnames(G)=="cCF" ), "aCF"]*sqrt(as.numeri
#
# cCFQ
#
# aCCQ <- male_i*B[which(colnames(G)=="aCC")]*male_Ca[which( colnames(G)=="aCC" ), "aCF"]*sqrt(as.numeri
#
# aCCQ
#
# cCCQ <- male_i*B[which(colnames(G)=="cCC")]*male_Ca[which( colnames(G)=="cCC" ), "aCF"]*sqrt(as.numeri
#
# cCCQ
#
# CCaQ <- (male_i*as.numeric(t(CCaB)%*%male_Ca[, "aCF"])*sqrt(as.numeric(t(CCaB)%*%male_Ca%*%CCaB))+fema
#
# CCcQ <- (male_i*as.numeric(t(CCcB)%*%male_Ca[, "cCF"])*sqrt(as.numeric(t(CCcB)%*%male_Ca%*%CCcB))+fema
#
# CWaQ <- (male_i*as.numeric(t(CWaB)%*%male_Ca[, "aCF"])*sqrt(as.numeric(t(CWaB)%*%male_Ca%*%CWaB))+fema
#
# CWcQ <- (male_i*as.numeric(t(CWcB)%*%male_Ca[, "cCF"])*sqrt(as.numeric(t(CWcB)%*%male_Ca%*%CWcB))+fema

```



```

#
# single_trait <- data.frame(trait=c("CFa", "CFc", "CCa", "CCc", "Cwa", "Cwc"), Sfr=c(CFaQ, CFcQ, CCaQ, CCcQ, CwQ, CwcQ))
# sum(c(CFaQ, CFcQ, CCaQ, CCcQ, CwQ, CwcQ))

# library(ggplot2)
# x <-
#
# dataframe <- data.frame(names = c("Overall", rep(c("CFa", "CFc", "CCa", "CCc", "Cwa", "Cwc"), times=2)), Sfr=
#
# barplot <- ggplot(dataframe, aes(x=names, y=Sfr, fill=Strategy))
# barplot + geom_bar(stat="identity", position="dodge")
# rep(rep(c("CFa", "index"), each=6), times=2)

```

Plots

```

library(ggplot2)
# Plot aggregate genotype
#AN_Qoverall_index <- Qoverall_index
LM_Qoverall_index <- 2
AN_Qoverall_CF <- 2
LM_Qoverall_CF <- 4

#Aggregate_Genotype <- data.frame(Gain=c(AN_Qoverall_index, LM_Qoverall_index, AN_Qoverall_CF, LM_Qoverall_CF))

#Aggregate_Genotype <- ggplot(Aggregate_Genotype, aes(Breed, Gain, fill=Strategy))

#Aggregate_Genotype + geom_bar(stat="identity", position="dodge")

```