

Function__GeneticGain

Silvan

30 11 2018

Function for covariance matrix between true and estimated breeding values

```
## @title
## Asymptotic covariance matrices between true and estimated breeding values.
## @description
## This function creates two covariance matrices between true and estimated breeding values of a male and a female selection candidate.
## @param genetic_var_cov The genetic variance-covariance matrix between the involved traits in the population.
## @param residual_var_cov The genetic variance-covariance matrix between the involved traits in the population.
## @param male_offspring Number of the assumed offspring of the male selection candidate when selection is performed.
## @param female_offspring Number of the assumed offspring of the female selection candidate when selection is performed.
## @param proportion_calves Proportion of calves in the offspring.
## @param proportion_adults Proportion of adults in the offspring.
## @param male_proportionselected Proportion selected of the offspring of a male selection candidate
## @param female_proportionselected Proportion selected of the offspring of a female selection candidate
## @return Asymptotic Covariance Matrices between true and estimated breeding values
compute_asymptotic_covariance_matrices <- function(genetic_var_cov,
                                                  residual_var_cov,
                                                  offspring,
                                                  proportion_calves,
                                                  proportion_adults,
                                                  male_proportionselected,
                                                  female_proportionselected){

  # define number of traits
  n <- as.numeric(nrow(genetic_var_cov))

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

  # Compute the relationship matrix dependent on the number of offspring of the selection candidate.
  suppressPackageStartupMessages(library(pedigreeemm))
  fnumb <- female_offspring+1
  female_A <- pedigree(sire = c(NA,rep(NA,times = female_offspring)), dam =c(NA,rep(1,times = female_offspring)))
  # Compute inverse of relationship matrix.
  female_Ainv <- getAInv(female_A)

  # Compute Relationship Matrix Z
  # numbers of adults and calves in scenario
  male_number_adults <- floor(proportion_adults*male_offspring)[[1]]
  male_number_calves <- male_offspring-male_number_adults
  female_number_adults <- floor(proportion_adults*female_offspring)[[1]]
  female_number_calves <- female_offspring-female_number_adults

  # Kronecker
```

```

kronecker_calves <- cbind(c(1,0,0),c(0,0,0),c(0,1,0),c(0,0,0),c(0,0,1),c(0,0,0))
kronecker_adults <- cbind(c(0,0,0),c(1,0,0),c(0,0,0),c(0,1,0),c(0,0,0),c(0,0,1))
# male Z
male_calves <- diag(1,male_number_calves)%x%kronecker_calves
male_calves <- cbind(male_calves,matrix(0, nrow=nrow(male_calves),ncol=male_number_adults*n))
male_adults <- diag(male_number_adults)%x%kronecker_adults
male_adults <- cbind(matrix(0, nrow=nrow(male_adults),ncol=male_number_calves*n),male_adults)
male_Z <- rbind(male_calves, male_adults)
male_Z<-cbind(matrix(0, nrow=male_offspring*(n/2),ncol=n),male_Z)
# female Z
female_calves <- diag(1,female_number_calves)%x%kronecker_calves
female_calves <- cbind(female_calves,matrix(0, nrow=nrow(female_calves),ncol=female_number_adults*n))
female_adults <- diag(female_number_adults)%x%kronecker_adults
female_adults <- cbind(matrix(0, nrow=nrow(female_adults),ncol=female_number_calves*n),female_adults)
female_Z <- rbind(female_calves, female_adults)
female_Z<-cbind(matrix(0, nrow=female_offspring*(n/2),ncol=n),female_Z)

# Genetic Variance Covariance Matrix G
G <- genetic_var_cov

# Residual Variance Covariance Matrix R
residual_var_cov_calves <- residual_var_cov[c(1,3,5),c(1,3,5)]
residual_var_cov_adults <- residual_var_cov[c(2,4,6),c(2,4,6)]
# male
male_calves_kronecker_residual <- diag(male_number_calves) %x% residual_var_cov_calves
male_adults_kronecker_residual <- diag(male_number_adults) %x% residual_var_cov_adults
male_calves_kronecker_residual_extended <- cbind(male_calves_kronecker_residual, matrix(0,nrow=nrow(male_calves_kronecker_residual),ncol=male_number_adults*n))
male_adults_kronecker_residual_extended <- cbind( matrix(0,nrow=nrow(male_adults_kronecker_residual),ncol=male_number_calves*n),male_adults_kronecker_residual)
male_R <- rbind(male_calves_kronecker_residual_extended,male_adults_kronecker_residual_extended)
# female
female_calves_kronecker_residual <- diag(female_number_calves) %x% residual_var_cov_calves
female_adults_kronecker_residual <- diag(female_number_adults) %x% residual_var_cov_adults
female_calves_kronecker_residual_extended <- cbind(female_calves_kronecker_residual, matrix(0,nrow=nrow(female_calves_kronecker_residual),ncol=female_number_adults*n))
female_adults_kronecker_residual_extended <- cbind( matrix(0,nrow=nrow(female_adults_kronecker_residual),ncol=female_number_calves*n),female_adults_kronecker_residual)
female_R <- rbind(female_calves_kronecker_residual_extended,female_adults_kronecker_residual_extended)

# Selection intensity i and value of truncation x
# 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)

# Factor of variance reduction k
# male
male_k <- male_i*(male_i-male_x)
male_k
# female
female_k <- female_i*(female_i-female_x)
female_k

# Prediction error variance matrix

```

```

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

# Asymptotic Covariance Matrix
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)

# Return
return(Ca)
}

```

Function for genetic gain for index in Sfr./year

```

#' @title
#' Genetic Gain in Sfr. / year for index.
#' @description
#' This function computes the genetic gain in Sfr. / year of the selection on the index. In this index
#' @param genetic_var_cov The genetic variance-covariance matrix between the involved traits in the a
#' @param residual_var_cov The genetic variance-covariance matrix between the involved traits in the a
#' @param male_offspring Number of the assumed offspring of the male selection candidate when select
#' @param female_offspring Number of the assumed offspring of the female selection candidate when se
#' @param proportion_calves Proportion of calves in the offspring.
#' @param proportion_adults Proportion of adults in the offspring.
#' @param male_proportionselected Proportion selected of the offspring of a male selection candidate
#' @param female_proportionselected Proportion selected of the offspring of a female selection candidat
#' @param male_generationintervall Generation intervall of the male selection path.
#' @param female_generationintervall Generation intervall of the female selection path.
#' @param economic_weights A vector of economic weights with lenght of number of traits.
#' @return Genetic Gain in Sfr. per year
compute_genetic_gain_overall_index <- function(genetic_var_cov,
                                              residual_var_cov,
                                              offspring,
                                              proportion_calves,
                                              proportion_adults,
                                              male_proportionselected,
                                              female_proportionselected,
                                              male_generationintervall,
                                              female_generationintervall,
                                              economic_weights){

  # define vector of economic weights
  a <- economic_weights

  # Intesity of selection
  male_i <- dnorm(qnorm(1-male_proportionselected))/male_proportionselected
  female_i <- dnorm(qnorm(1-female_proportionselected))/female_proportionselected

```

```

# Asymptotic covariance matrices

Ca <- compute_asymptotic_covariance_matrices(genetic_var_cov = genetic_var_cov, residual_var_cov = residual_var_cov)

# Number of traits
n <- as.numeric(ncol(Ca))

# male
male_Ca <- Ca[1:n,]
# female
female_Ca <- Ca[(1+n):(2*n),]

# Genetic Gain
Qoverall_index <- (male_i*sqrt(as.numeric(t(a)%*%male_Ca%*%a))+female_i*sqrt(as.numeric(t(a)%*%female_Ca%*%a)))

return(Qoverall_index)
}

```

Function for correlated genetic gain for index in trait unit / year

```

#' @title
#' Correlated genetic gain in trait unit / year for index.
#' @description
#' This function computes the correlated genetic gain in trait unit / year of the selection on the index.
#' @param genetic_var_cov The genetic variance-covariance matrix between the involved traits in the a.
#' @param residual_var_cov The genetic variance-covariance matrix between the involved traits in the a.
#' @param male_offspring Number of the assumed offspring of the male selection candidate when select
#' @param female_offspring Number of the assumed offspring of the female selection candidate when se
#' @param proportion_calves Proportion of calves in the offspring.
#' @param proportion_adults Proportion of adults in the offspring.
#' @param male_proportionselected Proportion selected of the offspring of a male selection candidate
#' @param female_proportionselected Proportion selected of the offspring of a female selection candidat
#' @param male_generationintervall Generation intervall of the male selection path.
#' @param female_generationintervall Generation intervall of the female selection path.
#' @param economic_weights A vector of economic weights with lenght of number of traits.
#' @param trait The name of the trait, of which the genetic gain should be.
#' @return Correlated genetic gain in trait unit per year
compute_genetic_gain_trait <- function(genetic_var_cov,
                                     residual_var_cov,
                                     offspring,
                                     proportion_calves,
                                     proportion_adults,
                                     male_proportionselected,
                                     female_proportionselected,
                                     male_generationintervall,
                                     female_generationintervall,
                                     economic_weights, trait){

  # define vector of economic weights
  a <- economic_weights

  # Intesity of selection

```

```

male_i <- dnorm(qnorm(1-male_proportionselected))/male_proportionselected
female_i <- dnorm(qnorm(1-female_proportionselected))/female_proportionselected

# Asymptotic covariance matrices

Ca <- compute_asymptotic_covariance_matrices(genetic_var_cov = genetic_var_cov, residual_var_cov = residual_var_cov)

# Number of traits
n <- as.numeric(ncol(Ca))

# male
male_Ca <- Ca[1:n,]
# female
female_Ca <- Ca[(1+n):(2*n),]

# Correlated genetic gain
Qtrait <- (male_i*as.numeric(t(a)%*%male_Ca[,trait])*1/(sqrt(as.numeric(t(a)%*%male_Ca%*%a))) +
          female_i*as.numeric(t(a)%*%female_Ca[,trait]) * 1/(sqrt(as.numeric(t(a)%*%female_Ca%*%a))))

return(Qtrait)
}

```

Produce result of function for covariance matrix

```

# 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)

# input offspring
male_offspring <- 20
female_offspring <- 5
proportion_adults <- 0.5
proportion_calves <- 0.5

# proportion selected
male_proportionselected <- 0.1
female_proportionselected <- 0.5

compute_asymptotic_covariance_matrices(genetic_var_cov = genetic_var_cov, residual_var_cov = residual_var_cov)

## Warning: package 'lme4' was built under R version 3.4.4

## 12 x 6 Matrix of class "dgeMatrix"
##           cCC      aCC      cCF      aCF      cCW
## [1,] 0.187207076 0.175503339 -0.015026731 -0.017127229 0.78244165
## [2,] 0.175503339 0.192352721 -0.003111346 0.010741070 0.62266224
## [3,] -0.015026731 -0.003111346 0.046780354 0.044805026 -0.16135631

```

```
## [4,] -0.017127229 0.010741070 0.044805026 0.055084440 -0.21030228
## [5,] 0.782441648 0.622662239 -0.161356311 -0.210302281 10.98959612
## [6,] 1.219475432 1.411164565 0.158193116 0.403643288 22.03388249
## [7,] 0.070089408 0.065326393 -0.008898659 -0.008705964 0.26871299
## [8,] 0.065326393 0.062455164 -0.004846760 -0.004393764 0.21844763
## [9,] -0.008898659 -0.004846760 0.012818838 0.012450411 -0.09243105
## [10,] -0.008705964 -0.004393764 0.012450411 0.012297442 -0.09763299
## [11,] 0.268712990 0.218447628 -0.092431049 -0.097632987 2.65858897
## [12,] 0.371471778 0.318829171 -0.053505802 -0.064357351 4.47043347
##
##      aCW
## [1,] 1.21947543
## [2,] 1.41116456
## [3,] 0.15819312
## [4,] 0.40364329
## [5,] 22.03388249
## [6,] 72.88438136
## [7,] 0.37147178
## [8,] 0.31882917
## [9,] -0.05350580
## [10,] -0.06435735
## [11,] 4.47043347
## [12,] 8.81889354
```

Produce result of function for genetic gain for index

```
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

# 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)

# input offspring
male_offspring <- 20
female_offspring <- 5
proportion_adults <- 0.5
proportion_calves <- 0.5
```

```

# proportion selected
male_proportionselected <- 0.1
female_proportionselected <- 0.5

# economic weight
economic_weights <- "LM_economic_values"
economic_weights <- read.table(file = economic_weights, sep=";", header=TRUE)

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

economic_weights <- as.matrix(economic_weights)
economic_weights <- as.vector(economic_weights)

# Generation Intervall
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)
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)

compute_genetic_gain_overall_index(genetic_var_cov = genetic_var_cov, residual_var_cov = residual_var_cov)

## [1] 0.1009993

```

Produce results of function for correlated genetic gains for index

```

library(dplyr)
# 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)
list_of_traits <- colnames(genetic_var_cov)
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)

# input offspring
male_offspring <- 20

```

```

female_offspring <- 5
proportion_adults <- 0.5
proportion_calves <- 0.5

# proportion selected
male_proportionselected <- 0.1
female_proportionselected <- 0.5

# economic weight
economic_weights <- "LM_economic_values"
economic_weights <- read.table(file = economic_weights, sep=";", header=TRUE)

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

economic_weights <- as.matrix(economic_weights)
economic_weights <- as.vector(economic_weights)

# Generation Intervall
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)
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)

# Loop that produces a matrix with all correlated genetic gains
mat_correlated_gen_gain <- NULL

for(trait in list_of_traits){
  if (is.null(mat_correlated_gen_gain)){
    mat_correlated_gen_gain <- cbind(trait,compute_genetic_gain_trait(genetic_var_cov = genetic_var_cov,
  } else {
    mat_correlated_gen_gain <- rbind(mat_correlated_gen_gain,cbind(trait,compute_genetic_gain_trait(genetic_var_cov = genetic_var_cov,
  })
}
mat_correlated_gen_gain

##      trait
## [1,] "cCC" "-0.0131988870528898"
## [2,] "aCC" "-0.000725854527595375"
## [3,] "cCF" "0.0204302248157807"
## [4,] "aCF" "0.0234180578615821"
## [5,] "cCW" "-0.542587354222553"

```



```
## [6,] "aCW" "-0.985309367836364"
```