Practical 3: Estimation of Genetic Parameters

Time schedule of practical session 3:

11:15	Question to lectures, multiple-choice question and follow up on previous multiple choice questions
11:30	Today´s exercise and assignment to groups
12:00	15 minutes break
12:30	Go through excercises using final word
12:50	Repeat multiple-choice questions
13:00	End of practical session 3

Introduction:

In this practical we will estimate genetic parameters (heritability) for quantitative traits observed in the F2 mouse population. We will be using the REML method. This method allow for estimation of genetic parameters using phenotypic information for individuals from a general pedigree. REML is based on linear mixed model methodology and uses a likelihood approach to estimate genetic parameters. The REML method also require us to calculate an genetic relationship matrix using a recursive algorithm. These methods and algorithms are implemented in the R package `qgg`.

This package provides an infrastructure for efficient processing of large-scale genetic and phenotypic data including core functions for:

- fitting linear mixed models
- constructing genetic relationship matrices
- estimating genetic parameters (heritability and correlation)
- performing genomic prediction and genetic risk profiling
- single or multi-marker association analyses

We will also be using the qgg package for the remaining practicals.

Installation of the R package qgg:

You can install qgg from CRAN with:

```
install.packages("qgg")
```

You can install the latest version of qgg from github with:

```
#install.packages("devtools") # needed if devtools is not allready installed
library(devtools)
options(devtools.install.args=" --no-multiargs")
devtools::install_github("psoerensen/qgg")
```

Load R packages that will be used in this practical

```
library(qgg) # R package used for REML analysis
#install.packages("corrplot")
library(corrplot)
```

Explore mouse pedigree data

The mouse data set can be loaded using the following command:

```
mouse <- readRDS(url("https://github.com/psoerensen/bgcourse/raw/main/data/mouseqtl.rds"))
```

The mouse pedigree is loaded in a similar way using the following command:

```
pedigree <- readRDS(url("https://github.com/psoerensen/bgcourse/raw/main/data/pedigree.rds"))
```

Question 1: Which variables do we have in the pedigree? Use the `str` function to get a fast overview of the pedigree you are working.

Answer:

Question 2: How many individuals do we have in the pedigree?

Answer:

Question 3: How many generations and number of mice in each generation do we have in the pedigree? Use the `table` function on the generation variable.

Answer:

Computing genetic relationship matrix for the mouse pedigree:

The REML analysis require us to calculate the genetic relationship matrix A. This is done using information about the id, mother, and father which is avaliable in our pedigree data file.

To illustrate this step we will first calculate it for a small part of the mouse pedigree. We are given the following pedigree and we want to compute the matrix A.

```
family <- c(13,14,84,1244,1248)
pedigree[family,]
```

```
##         id sire dam family    sex generation
## 13      13    0   0    0/0   Male         IC
## 14      14    0   0    0/0 Female         M6
## 84      84   13   14   13/14 Female          F1
## 1244 1244   78   84   78/84 Female          F2
## 1248 1248   78   84   78/84   Male          F2
```

The additive genetic relationship (A_{ij}) between the various sources (j) and the individual itself, i.e. the candidate to be evaluated (i), can be seen in the table below.

Relative	A_{ij}
Self	1.0
Unrelated	0
Mother	0.5
Father	0.5
Grandparent	0.25
Half-sib	0.25
Full-sib	0.5
Progeny	0.5

Answer:

Next we will compute the genetic relationship matrix for the entire mouse pedigree. The matrix A can be computed using a recursive algorithm implemented in the function `grm` from the qgg package. Use the command below to compute the genetic relationship matrix for the mouse pedigree:

```
A <- grm(pedigree=pedigree)
```

Question 4: What is the dimension of the genetic relationship matrix?

Answer:

The number of rows and columns should be equal to the number of individuals in the pedigree. Check the first 5 individuals in the matrix using the following command:

```
A[1:5,1:5]
```

```
##   1 2 3 4 5
## 1 1 0 0 0 0
## 2 0 1 0 0 0
## 3 0 0 1 0 0
## 4 0 0 0 1 0
## 5 0 0 0 0 1
```

Question 5: Are these individuals related?

Answer:

To further explore the genetic relationship we compute the mean of diagonal elements of A using the following command:

```
mean(diag(A))
```

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

Question 6: How should we interpret this value?

Answer:

Previously we have determined the genetic relationship matrix for a small part of the mouse pedigree. We can extract the corresponding elements from the A matrix for the entire mouse pedigree using the following command:

```
ids <- c(13,14,84,1244,1248)
A[ids,ids]
```

```
##        13   14  84 1244 1248
## 13   1.00 0.00 0.5 0.25 0.25
## 14   0.00 1.00 0.5 0.25 0.25
## 84   0.50 0.50 1.0 0.50 0.50
## 1244 0.25 0.25 0.5 1.00 0.50
## 1248 0.25 0.25 0.5 0.50 1.00
```

Question 6: Are the values in this part of the genetic relationship matrix the same as you have found using the "manual" approach?

Answer:

Make a plot of the genetic relationship matrix using the `corrplot` function from the corrplot R package:

```
corrplot(A, method="color", bg="white",
        outline=FALSE, col=NULL, tl.pos="n",is.corr = FALSE, xlab=FALSE, ylab=FALSE)
```

Question 7: Describe the plot you just made of the genetic relationship?

Answer:

Specifying the linear mixed model for the mouse data:

The next step is to prepare the linear mixed model for the mouse data. Recall that the linear mixed model contains the observation vector for the trait(s) of interest (y), the `fixed effects` that explain systematic differences in y, and the `random genetic effects` a and random residual effects e.

A matrix formulation of a general model equation is:

$$y = Xb + a + e$$

where

$$\begin{aligned} y &: \text{is the vector of observed values of the trait,} \\ b &: \text{is a vector of fixed effects,} \\ a &: \text{is a vector of random genetic effects,} \\ e &: \text{is a vector of random residual effects,} \\ X &: \text{is a known design matrix that relates the elements of b to their corresponding element in y.} \end{aligned}$$

In the statistical model (specified above) the random effects (a and e) and the phenotypes (y) are considered to be random variables which follow a multivariate normal distribution: In general terms the expectations of these random variables are:

$$\begin{aligned} E(y) &= Xb \\ E(a) &= 0 \\ E(e) &= 0 \end{aligned} \tag{1}$$

and the variance-covariance matrices are:

$$\begin{aligned} Var(a) &= A\sigma_a^2 \\ Var(e) &= I\sigma_e^2 \\ Var(y) &= A\sigma_a^2 + I\sigma_e^2 \end{aligned}$$

where $A\sigma_a^2$, and $I\sigma_e^2$ are square matrices of genetic and residual (co)variances among the individuals, respectively. In the previous section we have allready constructed the genetic relationship matrix A.

In order to perform the REML analysis we need to construct y and X from the mouse data. Let us just have a quick look at the mouse data again:

```
str(mouse)
```

```
## ’data.frame':    1177 obs. of  8 variables:
##  $ sire : Factor w/ 11 levels "25","28","34",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ dam  : Factor w/ 55 levels "26","27","29",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ sex  : Factor w/ 2 levels "Female","Male": 1 1 1 1 2 2 2 2 1 1 ...
##  $ reps : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 2 2 ...
##  $ Gl   : num  187 136 115 125 112 190 169 159 111 89 ...
##  $ BW   : num  36.6 33.3 42.1 37.1 38.4 ...
##  $ M227 : Factor w/ 3 levels "AA","AB","BB": 2 1 2 2 2 1 2 1 2 2 ...
##  $ M1139: Factor w/ 3 levels "AA","AB","BB": 3 NA 1 1 1 2 3 3 2 2 ...
```

Here we will estimate the heritability for body weight. The vector of observed trait values for body weight can be extracted from the mouse data as follows:

```
y <- mouse[,"BW"]
```

Let us explore the trait values using the `head`, `tail` and `summary` functions:

```
head(y)
```

```
## [1] 36.65 33.29 42.07 37.15 38.39 39.82
```

```
tail(y)
```

```
## [1] 39.67 39.35 44.80 52.23 47.63 54.10
```

```
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   23.04   34.06   38.32   38.72   43.40   60.28
```

To make the X matrix we need to decide which variables we should include as fixed effects in the model. We have sex, reps, sire, dam, M227 and M1139 in the mouse data frame.

Question 8: Which variables should we include as fixed effects in the model?

Answer:

The `model.matrix` function can be used to construct the X matrix in the linear mixed model specified above:

```
X <- model.matrix(BW ~ sex + reps, data=mouse)
```

We can use the `head` and `tail` functions to look at the X matrix:

```
head(X)
```

```
##    (Intercept) sexMale reps2 reps3
## 91           1       0     0     0
## 92           1       0     0     0
## 93           1       0     0     0
## 94           1       0     0     0
## 95           1       1     0     0
## 96           1       1     0     0
```

```
tail(X)
```

```
##      (Intercept) sexMale reps2 reps3
## 1262           1       0     0     1
## 1263           1       0     0     1
## 1264           1       1     0     1
## 1265           1       1     0     1
## 1266           1       1     0     1
## 1267           1       1     0     1
```

Estimating genetic parameters on the mouse data using REML:

The goal of the REML analysis to estimate the parameters (i.e. variance components σ_a^2 and σ_e^2) in the linear mixed model specified above. In this analysis we find the set of parameters which maximizes the **likelihood** of the data, i.e., the probability of observations given the model and its parameter estimates: $p(y|\hat{b}, \hat{\sigma}_a^2, \hat{\sigma}_e^2)$.

The input required the vector of observed values of the trait (y), the deisgn matrix for the fixed effects (X), and the genetic relationship matrix (A). The A matrix calculated previously include genetic relationships for all individuals in the pedigree. However only a subset of the inviduals have phenotypes recorded for body weight. Therefore we need to subset the A matrix as shown in the R code below:

```
ids <- rownames(X)
A <- A[ids,ids]
```

The REML method is implemented in the `greml` function from the "qgg" package. The REML analysis is done using the following command:

```
fit <- greml(y=y,X=X, GRM=list(A=A))
```

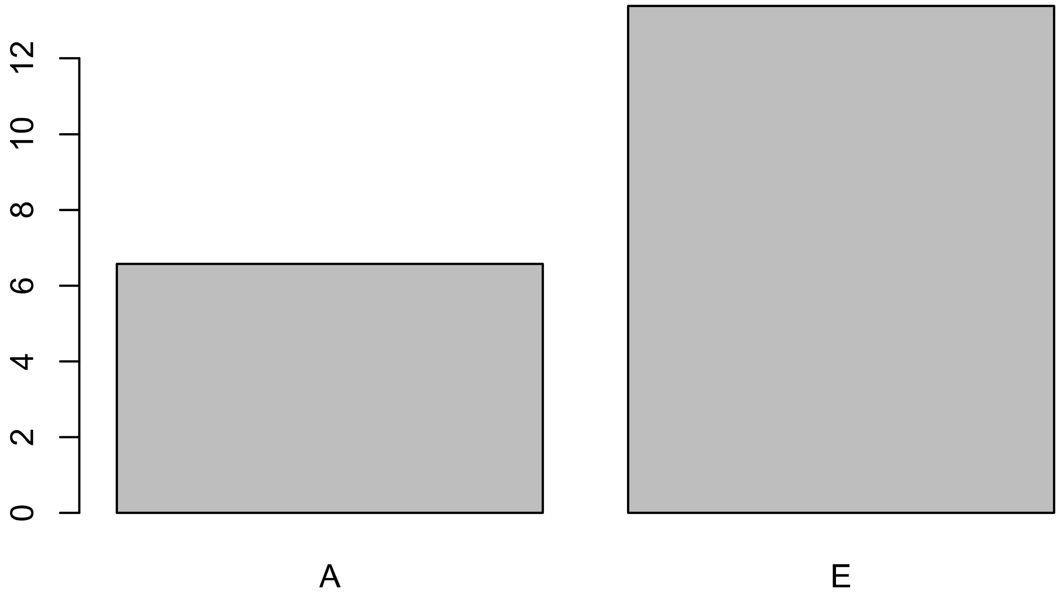
The fit object (i.e., output from the `greml` function) contains estimates of variance components, fixed and random effects, first and second derivatives of log-likelihood, and the asymptotic standard deviation of parameter estimates.

Our main interest is the variance components σ_a^2 and σ_e^2 which are in the `fit$theta` slot of the fit. The following commands extract and makes a barplot of the estimates of the variance components:

```
fit$theta
```

```
##         A         E
##  6.569611 13.384147
```

```
barplot(fit$theta)
```



The first element in the **theta** vector is the estimate of the additive genetic variance ($\hat{\sigma}_a^2$) and the second element is the estimate of the residual variance ($\hat{\sigma}_e^2$).

```
Va <- fit$theta[1]
Ve <- fit$theta[2]
Va
```

```
##        A
## 6.569611
```

```
Ve
```

```
##        E
## 13.38415
```

From the REML estimate of the variance components, the heritability can easily be computed by:

$$\hat{h}^2 = \hat{\sigma}_a^2/(\hat{\sigma}_a^2 + \hat{\sigma}_e^2) \tag{2}$$

where the hat (ˆ) refers to estimators.

Question 9: What is the heritability for body weight?

Answer:

In the experiment the mice were feed ad libitum. Now we want to perform a simlar experiment where mice are reared under restricted feed intake, We will record phenotypes for body weight and blood glucose levels and use mice from the same F2 population.

Question 10: Should we re-estimate the heritability?

Answer:

Question 11: What is the heritability for glucose levels in the blood?

Answer:

```r
y <- mouse[,"Gl"]
X <- model.matrix(Gl ~ sex + reps, data=mouse)
ids <- rownames(X)
A <- grm(pedigree=pedigree)
A <- A[ids,ids]
fit <- greml(y=y,X=X, GRM=list(A=A))
Va <- fit$theta[1]
Ve <- fit$theta[2]
Va/(Va+Ve)
```

```
##         A
## 0.4058474
```