The R-Package 'synbreed'

Valentin Wimmer*

September 23, 2010

Abstract

This document gives an introduction to the R-package 'synbreed' which contains tools and methods for plant and animal breeding. The goal is the creation of an analysis pipeline for genomic selection. This comprises tools for genotypic, phenotypic and pedigree data. The steps of a typical analysis are presented in this document. This starts with the coding of the marker data,followed by the construction of relationship matrices according to pedigree or genomic relationship matrices based on marker data, e.g. according to vanRaden (2008). At the end the use of linear mixed models for genomic selection is described.

Keywords: synergistic plant and animal breeding, kinship, pedigree, genomic marker data, mixed models, genomic selection

1 Introduction

The R-package 'synbreed' aims to provide the tools that are necessary to analyze data of breeding programs and perform genomic selection. Of course, there exisits already software for this purpose. The idea of this package is to collect the methods in one package, so that analysis can be performed in one software with just a few steps as described in this document. Additional, this package takes care of special problems of modern breeding programs as the use of doubled haploid (DH) lines in plant breeding. Most of packages source code is written in R, so that methods could easily be adopted for special purposes.

Modern breeding programs make use available genomic information of individuals. On the genomic level, individuals could be distinguished by *alleles* which are different forms of a particular gene. In diploid species, every individuals has two sets of chromosomes and thus two copies of each allele. If both alleles are the same, the individual is homozygous for this

*Author of correspondence. Contact: Institute for plant breeding, Technical University of Munich, Emil-Ramann-Str. 4, 85354 Freising, Germany, Email: `Valentin.Wimmer@wzw.tum.de`

gene, otherwise it heterozygous. *genomic markers* are used to identify differences between individuals for specific loci on the genome. So called SNP (single nucleotide polymorphism) markers detect variation occurring when a single nucleotide (A, T, C, or G) at the same loci differ between individuals. In this document, the term *genotype* refers to an individual's set of genes (and alleles) and is used as a synonym for an individual. On the other hat, *phenotype* denotes the observed and measured value of an genotype, i.e. a trait of commercial interst in breeding programs. It is assumed that the phenotype is determined by a certain degree the genotype and by the environment.

The idea of *genomic selection* is that some of the markers are linked to a *quantitative trait loci* (QTL) related to a trait. Meuwissen et al. (2001) proposed to make use of the markers and regress the phenotype on the markers (genotype). This model assigns an effect to each of the markers. The estimated breeding value of a genotype consists of the sum of all makrer effects. Thus in genomic selection individuals with a favorable set of genes are selected the for the next cycle in breeding scheme.

The remainder of this document is structured as follows. In section 2 a simulated data set is presented which is used to illustrate the methods in this document. Section 3 describes the formatting of marker data and section 5 shows how to utilize pedigree information. Section 4 shows how Linkage Disequilibrium between markers could be exploited using synbreed package. Section 6 presents several methods to set up expected and realized relationship matrices.

2 Example data

In this document the steps of an analysis pipeline for genotypic and phenotypic data in plant or animal breeding with the R-package 'synbreed' are presented. For illustration, a the package contains a simulated data set for maize, called maize. This data set could be used to test performance of methods because estimaed values could easily be compared with specified parameters of the simulation (position of QTL, size of marker effects, true breeding values for individuals). To load maize data, use

```
> library(synbreed)
> data(maize)
```

This data set contains genotypic and phenotypic data, as well as pedigree information up to grandparents for 1250 maize lines. When loading maize data, in fact the following four data sets are loaded into workspace

maize.geno This is a data.frame containing the marker data of 696 biallelic SNP-markers for the 1250 genotypes. The first column of the data set contains the ID for the identification of the genotypes. This variable should be used for the merge with the phenotypic data. The marker data is coded with 0/1 and because of simulated data no missing values are present. Note that the coding does not contain any information about allele frequencies, thus 1 could be minor or major allele.

maize.pheno This is a `data.frame` has one column `ID` and a column `Trait` containing the measured phenotypic trait. The order of the genotypes is the same as the order of rows in `maize.geno`.

maize.ped This `data.frame` contains the pedigree information of 1301 genotypes (1250 lines and 51 ancestors).

maize.marker.pos This `data.frame` contains additional information for the SNP markers. The first column `pos` gives the position of the marker on the chromosome in cM. The second column `chr` sepecifies the chromosome (linkage group) the marker belongs to. The order of the markers are the same as the order of columns in `maize.geno`.

3 Marker data

In the first step, marker data has to be coded in a way that it could be used for the construction of genomic relationship matrices. For biallelic marker data, the minor allele should be coded as 2 and the major allele as 0. This task is done by the function `codeGeno`. If no missing values and no heterozygous genotypes for any loci are present and all markers should be used in the following analyses, this function does simply recode the alleles as mentioned above. For the `maize` data, use

```
> marker <- codeGeno(maize.geno[, -1])
```

to obtain an object `marker` which contains the recoded marker data. Note that the first column is not used because it contains the `ID`. Now, the minor allele frequencies are easily obtained by dividing the column means of `marker` by 2. A histogram of minor allele frequencies is shown in Figure 1.

In experimental data usually missing values occur in genotypic data due to different reasons (i.e. heterozygous genotypes for one locus in homozygous lines). The function `codeGeno` provides the possibility to impute missing values by chance or according to family structure using the following rules:

with population structure Suppose an observation i is missing (NA) for a marker j in population k. If marker j is fixed in population k, the imputed value will be the fixed allele. If marker j is segregating in population k, the value is 0 with probability 0.5 and 2 with probability 0.5.

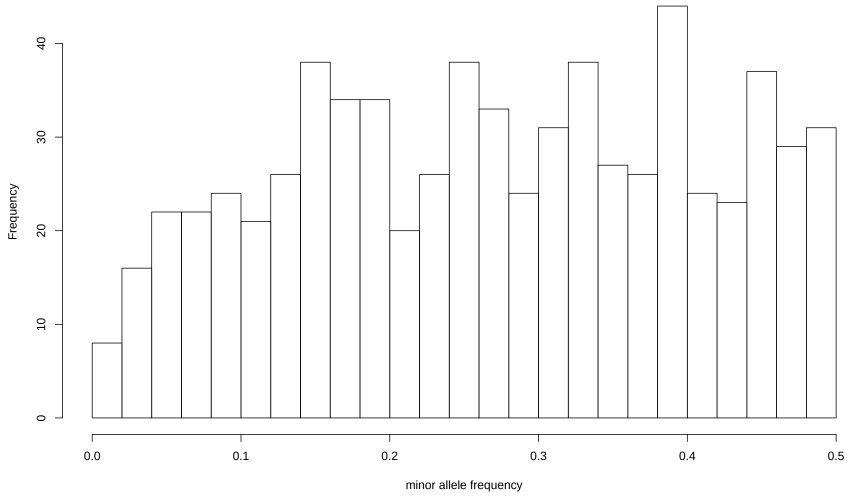


Figure 1: Histogram of the minor allele frequency of the 696 SNP markers in `maize` data.

without population structure The missing values for a marker j are sampled from the allele distribution of marker j.

To illustrate the difference in classification of missing values, 200 entries of the marker matrix are selected, the values saved and these entries are coded as `NA`.

```
> marker <- as.matrix(marker)
> ind1 <- sample(1:nrow(marker), 200)
> ind2 <- sample(1:ncol(marker), 200)
> posNA <- cbind(ind1, ind2)
> original <- marker[posNA]
> marker[posNA] <- NA
```

The number of 1250 genotypes in the `maize` data consist of 25 *half sib* families with 50 genotypes in each family. The genotypes are ordered according to the family structure.

```
> pop <- rep(1:25, each = 50)
```

Recoding of the marker data and imputing of the missing values is done as follows

```
> marker1 <- codeGeno(marker, impute = TRUE, pop)


approximative run time  0  seconds
 ...
```

```
total number of missing values                 : 200
number of imputations by family structure      : 123
number of random imputations                   : 77
approximate fraction of correct imputations : 0.808
```

A report is printed on the screen which informs about the number of imputations performed according to family structure n_F or chance n_R. The approximate fraction of correct imputations is $\frac{n_F+0.5n_R}{n_F+n_R}$. For the simulated data the original values are known. With the following commands the quality of the classification of the missing values is judged:

```
> imputed <- marker1[posNA]
> (t1 <- table(original, imputed))
```

```
        imputed
original   0   2
       0 128  18
       2  24  30
```

The fraction of correct replacements is

```
> sum(diag(t1))/sum(t1)
```

```
[1] 0.79
```

Note that expected fraction of correct imputation without family structure equals 0.5.

In an analysis of genotypic data it is common to discard marker with a small minor allele frequency and/or many missing values. There are two additional arguments `maf` and `nmiss` for function `codeGeno`. Before recoding the data all marker with more than `nmiss`$\cdot$100% missing values are discarded. After recoding the maker data only markers with a minor allele frequency $>$ `maf` are returned by the function. By default, no markers are selected by one of both criteria, thus `maf`=`nmiss`=0.

Note that missing values in the marker data must be coded as `NA`. Instead of imputing the values `codeGeno` provides the possibility to replace the missing values by a certain value, i.e. 1 which is the expectation. Different codings of the alleles could easily be obtained by linear transformations of the marker matrix.

4 Linkage Disequilibrium

Linkage Disequilibrium (LD) is defined as a non-random association between polymorphisms at two or more loci. It is calculated as the difference between observed and expected (assuming random distributions) allelic frequencies.

There are many possibilities to compute LD from genotypic data. In synbreed package, LD between two loci i and j denoted as LD_{ij} is computed as coefficient of determination R^2 between the data of the genotypes $\mathbf{x_i}$ and $\mathbf{x_j}$ at both loci. $\mathbf{x_i}$ is an n-dimensional vector containing marker data of n individuals. This equals squared correlation coefficient of both data vectors, thus

$$LD_{ij} = r(\mathbf{x_i}, \mathbf{x_j})^{\mathbf{2}}.$$

Usually the overall amount of LD of markers in-between each linkage groups is of interest as well as the decline of LD when physical distance of markers is increasing. To plot the distance versus the LD for each linkage group, use function `LDDist`, to make a LD Heatmap for values of R^2, use function `LDMap`. For `maize` we plot the LD for the first chromosome

```
> data(maize)
> marker <- codeGeno(maize.geno[, -1])
> chr1 <- maize.marker.pos$chr == 1
> LDDist(marker[, chr1], maize.marker.pos$chr[chr1], maize.marker.pos$pos[chr1])
```

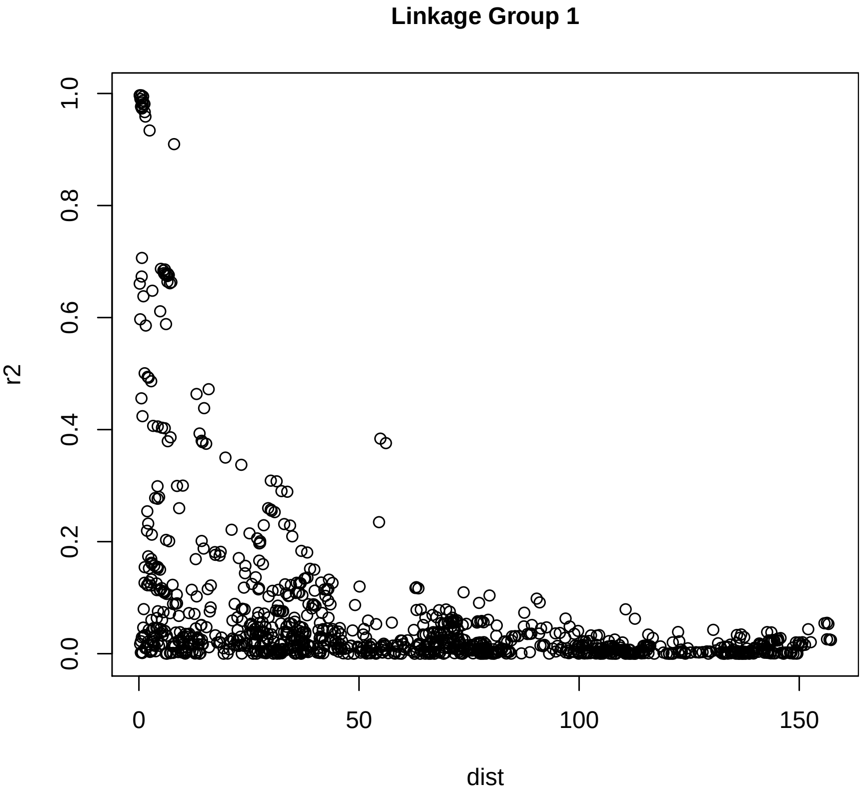


Figure 2: LD versus distance plot for first chromosome of `maize` data.

LD heatmap for pairwise LD of markers on the first chromosome is obtained as

```
> LDMap(marker[, chr1], maize.marker.pos$chr[chr1], maize.marker.pos$pos[chr1])
```

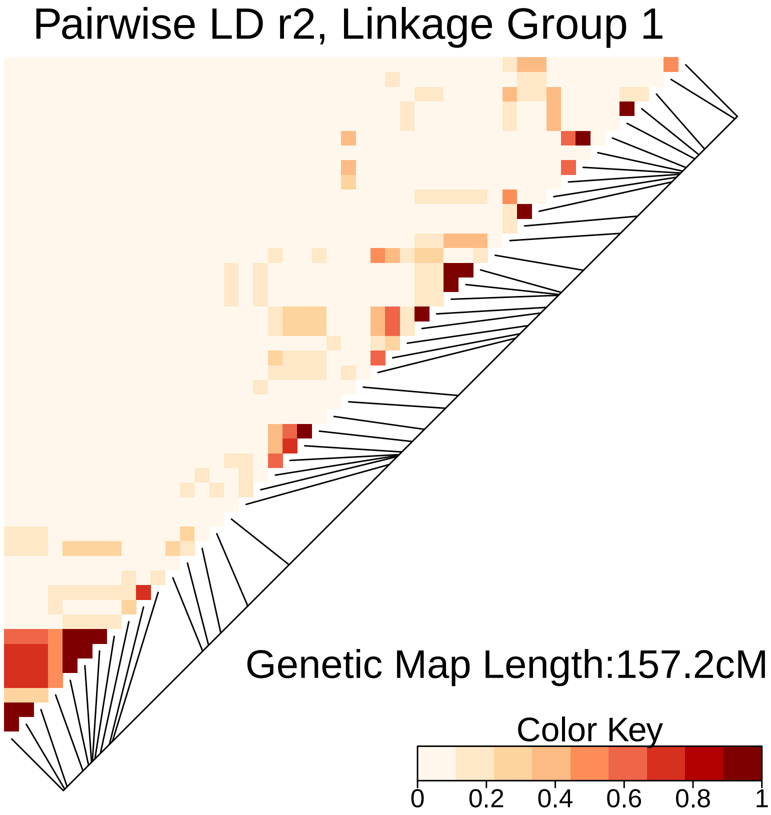


Figure 3: LD heatmap for markers on first chromosome of `maize` data.

5 Pedigree

An important source of information in breeding programs is pedigree information. Especially in animal breeding, pedigree is recorded over many generations. The pedigree usually consists of a list of individuals (animals or plants) of the current generation which is the subject of analysis and their ancestors (for which usually no phenotypic data is available). The pedigree is sorted the generation, beginning with the individuals with unknown parents. An example for a pedigree with five individuals belonging to 4 generations is given below.

Note that unknown parents are coded as ”0” in `synbreed` package and

ID	Par1	Par2	gener
A	-	-	0
B	-	-	0
C	A	B	1
D	A	C	2
E	D	B	3

generation starts with 0. In `synbreed` exists the class "pedigree", which should be used for handling pedigree information. An object of class "pedigree" consists of a `data.frame` with at least variables `ID`, `Par1`, `Par2` and `gener`. The function `create.pedigree` creates an object of class "pedigree" for a given set of individuals and the pair of parents. The generation can be specified by the user or optional computed by the function.

Suppose we have the pedigree structure of the example. This structure is carried into `synbreed` package with the following command:

```
> id <- c("A", "B", "C", "D", "E")
> par1 <- c(0, 0, "A", "A", "D")
> par2 <- c(0, 0, "B", "C", "B")
> ped <- create.pedigree(id, par1, par2)
> ped
```

```
  ID Par1 Par2 gener
1  A    0    0     0
2  B    0    0     0
3  C    A    B     1
4  D    A    C     2
5  E    D    B     3
```

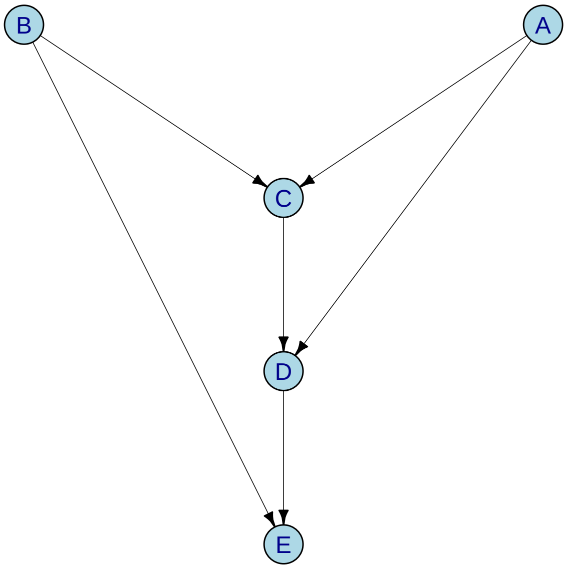
An object of class "pedigree" could be visualized with generic plotting function for S3 class "pedigree".

It is possible to simulate a pedigree structure with function `simul.pedigree`. As arguments, the number of generations to simulate and the number of individuals in each generation has to be specified. By default, random mating is assumed in each generation. As there are no further restrictions, it is possible that inbreeds could be generated when parent 1 equals parent 2. To simulate a pedigree with 6 generations and 4, 6, 7, 9, 10 and 10 individuals in each generation, use

```
> set.seed(123)
> ped.simul <- simul.pedigree(gener = 6, ids = c(4, 6,
+     7, 9, 10, 10))
```

The resulting pedigree is visualized in Figure 4.

```
> plot(ped)
```



6 Relationship matrices

Pedigree information is usually used to set up relationship matrices for a set of individuals. The relationship is constructed by the *expected* fraction of alleles that are identical by descent (IBD) between relatives. Another possibility to set up a relationship is the use of marker data to compute the genomic relationship. This gives the *observed* relationship of individuals.

6.1 Based on Pedigree

The computation of the pedigree based relationship in `synbreed` starts with the gametic relationship. A gamete is the genetic unit that an individual passes to its offspring. The genetic value of an individual at one locus consists of two alleles. Suppose there is an individual C with the parents A and B. Individual C has to alleles C1 and C2. The source of allele C1 is Parent A, thus allele C1 could either be IBD to A1 or A2. Allele C2 was inherited of parent B, thus it could be IBD to B1 or B2. To compute the

```
> plot(ped.simul)
```

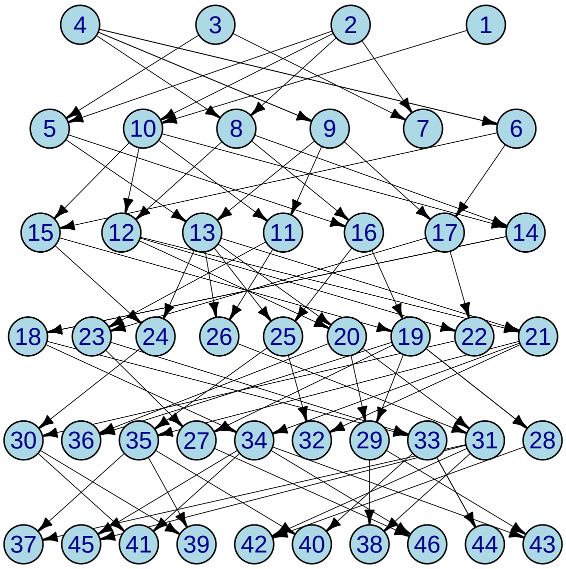


Figure 4: Simulated pedigree structure

gametic relationship start with an expanded table with two alleles for each individual.

ID	Allele	Par1	Par2
A	A1	-	-
A	A2	-	-
B	B1	-	-
B	B1	-	-
C	C1	A1	A2
C	C2	B1	B2
D	D1	A1	A2
D	D2	C1	C2
E	E1	D1	D2
E	E2	B1	B2

This table is converted into the gametic relationship **G** matrix which is of order $2n$, if the number of individuals is n. The diagonal values of **G** are always 1. The off-diagonal values give the probability that two alleles A_1 and A_2 are identical by descent (IBD), denoted as $P(A_1 \equiv A_2)$. If the parents are unknown, it is assumed that they are progeny of a random mating population. In this case the off-diagonals are zero. The gametic relationship matrix is constructed recursively, starting with the first generation in pedigree. The combination of $2^2 = 4$ alleles that describe the relationship of progeny A with parent C are computed as follows

$$\begin{aligned}
P(A_1 \equiv C_1) &= 0.5 \cdot [P(A_1 \equiv A_1) + P(A_1 \equiv A_2)] \\
P(A_1 \equiv C_2) &= 0.5 \cdot [P(A_1 \equiv B_1) + P(A_1 \equiv B_2)] \\
P(A_2 \equiv C_1) &= 0.5 \cdot [P(A_2 \equiv A_1) + P(A_2 \equiv A_2)] \\
P(A_2 \equiv C_2) &= 0.5 \cdot [P(A_2 \equiv B_1) + P(A_2 \equiv B_2)]
\end{aligned}$$

The gametic relationship for a given pedigree is obtained as follows

```
> G <- kinship(ped, ret = "gam")
> G
```

```
      A_1   A_2   B_1   B_2 C_1  C_2   D_1   D_2   E_1   E_2
A_1 1.000 0.000 0.000 0.000 0.5 0.00 0.500 0.250 0.375 0.000
A_2 0.000 1.000 0.000 0.000 0.5 0.00 0.500 0.250 0.375 0.000
B_1 0.000 0.000 1.000 0.000 0.0 0.50 0.000 0.250 0.125 0.500
B_2 0.000 0.000 0.000 1.000 0.0 0.50 0.000 0.250 0.125 0.500
C_1 0.500 0.500 0.000 0.000 1.0 0.00 0.500 0.500 0.500 0.000
C_2 0.000 0.000 0.500 0.500 0.0 1.00 0.000 0.500 0.250 0.500
D_1 0.500 0.500 0.000 0.000 0.5 0.00 1.000 0.250 0.625 0.000
D_2 0.250 0.250 0.250 0.250 0.5 0.50 0.250 1.000 0.625 0.250
E_1 0.375 0.375 0.125 0.125 0.5 0.25 0.625 0.625 1.000 0.125
E_2 0.000 0.000 0.500 0.500 0.0 0.50 0.000 0.250 0.125 1.000
attr(,"class")
[1] "relationshipMatrix"
```

The resulting object `G` is of class ”relationshipMatrix” which is the general class for all kinds of relationship matrices (gametic relationship, additive and dominance relationship, kinship). An object of class ”relationshipMatrix” is basically a symmetric matrix containing the relationship coefficient of two individuals. Note that the entry in `G` of allele x_1 and allele x_2 of an individual X equals his inbreeding coefficient

$$F_X = P(x_1 \equiv x_2).$$

For example, the inbreeding coefficent of individual D is

```
> as.numeric(G["D_1", "D_2"])
```

```
[1] 0.25
```

which is nonzero because individuals A and C, which are the parents of D, are relatives. Once the gametic relationship is computed, it could be converted in the additive numerator relationship matrix **A** or the dominance relationship matrix **D**. The additive relationship between individuals X and Y is given by

$$A_{XY} = \begin{cases} 1 + F_X, & X = Y \\ 2f_{XY}, & X \neq Y \end{cases}$$

where f_{XY} is the *coefficient of coancestry* (Wright, 1922) which is defined as

$$f_{XY} = P(X \equiv Y) = \frac{1}{4}\left[P(x_1 \equiv y_1) + P(x_1 \equiv y_2) + P(x_2 \equiv y_1) + P(x_2 \equiv y_2)\right]. \tag{1}$$

The additive numerator relationship matrix describes the additive relationship between individuals and is of order n. It is typically used in the animal model to estimate breeding values (additive genetic effects), see Section 7 The additive numerator relationship matrix for a given pedigree is obtained as follows

```
> A <- kinship(ped, ret = "add")
> A


      A     B     C    D     E
A 1.000 0.000 0.500 0.75 0.375
B 0.000 1.000 0.500 0.25 0.625
C 0.500 0.500 1.000 0.75 0.625
D 0.750 0.250 0.750 1.25 0.750
E 0.375 0.625 0.625 0.75 1.125
attr(,"class")
[1] "relationshipMatrix"
```

Note that the diagonals of **A** are $1 + F_i$ for $i = 1, ..., n$. Sometimes the kinship matrix is required, which is half of the additive numerator relationship matrix. It is obtained by

```
> K <- kinship(ped, ret = "kin")
```

Additionally it is possible to derive the dominance relationship matrix **D** out of **G**. The dominance is needed in the non-additive animal model. The dominance relationship between individuals X and Y is given by is defined as (Oakey et al., 2007)

$$D_{XY} = \begin{cases} 1 - F_X, & X = Y \\ t_{XY}, & X \neq Y \end{cases}$$

with t_{XY} being the coefficient of dominance covariance defined as

$$\begin{aligned} t_{XY} &= P(x_1 \equiv y_1 \neq x_2 \equiv y_2) + P(x_1 \equiv y_2 \neq x_2 \equiv y_1) \\ &= P(x_1 \neq x_2)P(y_1 \neq y_2)\left[P(x_1 \equiv y_1)P(x_2 \equiv y_2) + P(x_2 \equiv y_1)P(x_1 \equiv y_2)\right] \\ &= (1 - F_X)(1 - F_Y)\left[P(x_1 \equiv y_1)P(x_2 \equiv y_2) + P(x_2 \equiv y_1)P(x_1 \equiv y_2)\right]. \end{aligned}$$

Note that for a completely homozygous line X dominance is zero because $F_X = 1$. Dominance relationship matrix for the example pedigree is obtained by

```
> D <- kinship(ped, ret = "dom")
> D


       A        B       C         D         E
A 1.0000 0.000000 0.00000 0.1875000 0.0000000
B 0.0000 1.000000 0.00000 0.0000000 0.1093750
C 0.0000 0.000000 1.00000 0.1875000 0.2187500
D 0.1875 0.000000 0.18750 0.7500000 0.1025391
E 0.0000 0.109375 0.21875 0.1025391 0.8750000
attr(,"class")
[1] "relationshipMatrix"
```

Variance-covariance matrices for effects of higher order interactions in the non-additive animal model as additive-additive (AA), additive-dominance (AD) or dominance-dominance (DD) variance-covariance matrices can be computed as products of the variance-covariance matrices. Models for genomic selection can be extended by these effects, but the contribution of three-way or higher interactions is usually small (Bernardo, 2002).

6.2 Based on marker data

The relationship matrix based on marker data or genomic relationship matrix data represents the true relationship between relatives more precise than the numerator relationship based on pedigree, as it takes into account that relationship may deviate from the expected average relationship due to Mendelian sampling effect. Two methods for the construction of a relationship matrix based on marker data are implemented in the `synbreed` package: genomic relationship according to vanRaden (vanRaden, 2008) and according to Roger's distance (Rogers, 1972).

For vanRaden, the SNP genotypes are coded as the number of copies of one of the SNP alleles, i.e., 0, 1 or 2 (any linear transformations of these values are valid too). Thus the marker data could be the result of a call of `codeGeno` when imputing for the missing values was performed or the missing values were replaced with the value 1. The genomic relationship matrix according to vanRaden for n individuals and p marker is computed as

$$\frac{\mathbf{ZZ}'}{2\sum_{i=1}^{p} p_i(1-p_i)}, \tag{2}$$

where $\mathbf{Z} = \mathbf{M} - \mathbf{P}$ and $\mathbf{M}$ is the $n \times p$ marker matrix and $\mathbf{P}$ contains the allele frequencies multiplied by 2. p_i is the allele frequency of marker i. As an example we look at the marker data of 6 individuals genotyped with 8 SNP markers. Let

$$\mathbf{M} = \begin{pmatrix} 2 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 2 & 2 & 2 & 0 & 2 & 2 \\ 2 & 0 & 2 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 2 \end{pmatrix},$$

then it holds that

$$\mathbf{P} = \begin{pmatrix} 1.33 & 0.33 & 1.67 & 1.67 & 0.67 & 0 & 1.33 & 0.67 \\ 1.33 & 0.33 & 1.67 & 1.67 & 0.67 & 0 & 1.33 & 0.67 \\ 1.33 & 0.33 & 1.67 & 1.67 & 0.67 & 0 & 1.33 & 0.67 \\ 1.33 & 0.33 & 1.67 & 1.67 & 0.67 & 0 & 1.33 & 0.67 \\ 1.33 & 0.33 & 1.67 & 1.67 & 0.67 & 0 & 1.33 & 0.67 \\ 1.33 & 0.33 & 1.67 & 1.67 & 0.67 & 0 & 1.33 & 0.67 \end{pmatrix}$$

$$\mathbf{Z} = \begin{pmatrix} 0.67 & -0.33 & -1.67 & 0.33 & 1.33 & 0.00 & -1.33 & -0.67 \\ 0.67 & -0.33 & 0.33 & 0.33 & 1.33 & 0.00 & 0.67 & 1.33 \\ 0.67 & -0.33 & 0.33 & 0.33 & -0.67 & 0.00 & 0.67 & -0.67 \\ -1.33 & -0.33 & 0.33 & 0.33 & -0.67 & 0.00 & 0.67 & -0.67 \\ -1.33 & -0.33 & 0.33 & -1.67 & -0.67 & 0.00 & 0.67 & -0.67 \\ 0.67 & 1.67 & 0.33 & 0.33 & -0.67 & 0.00 & -1.33 & 1.33 \end{pmatrix}$$

and

$$\mathbf{ZZ}' = \begin{pmatrix} 7.44 & 0.11 & -1.22 & -2.56 & -3.22 & -0.56 \\ 0.11 & 4.78 & -0.56 & -1.89 & -2.56 & 0.11 \\ -1.22 & -0.56 & 2.11 & 0.78 & 0.11 & -1.22 \\ -2.56 & -1.89 & 0.78 & 3.44 & 2.78 & -2.56 \\ -3.22 & -2.56 & 0.11 & 2.78 & 6.11 & -3.22 \\ -0.56 & 0.11 & -1.22 & -2.56 & -3.22 & 7.44 \end{pmatrix}$$

with the denominator being $2\sum_{i=1}^{p} p_i(1-p_i) = 2.611$. To compute the genomic relationship according to vanRaden, matrix $\mathbf{M}$ is passed to the function `vanRaden`

```
> M <- matrix(data = c(2, 0, 0, 2, 2, 0, 0, 0, 2, 0, 2,
+      2, 2, 0, 2, 2, 2, 0, 2, 2, 0, 0, 2, 0, 0, 0, 2, 2,
+      0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0,
+      0, 0, 2), nrow = 6, byrow = TRUE)
> vR <- vanRaden(M)
> round(vR, 3)

       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  2.851  0.043 -0.468 -0.979 -1.234 -0.213
[2,]  0.043  1.830 -0.213 -0.723 -0.979  0.043
[3,] -0.468 -0.213  0.809  0.298  0.043 -0.468
[4,] -0.979 -0.723  0.298  1.319  1.064 -0.979
[5,] -1.234 -0.979  0.043  1.064  2.340 -1.234
[6,] -0.213  0.043 -0.468 -0.979 -1.234  2.851
attr(,"class")
[1] "relationshipMatrix"
```

Note the object `vR` is again of class ”relationshipMatrix”.

Another possibility is to compute the genomic relationship matrix according to Roger’s distance. Roger’s distance is computed as

$$d = \frac{1}{p}\sum_{i=1}^{p}\sqrt{1/2\sum_{j=1}^{n_i}(p_{ij}-q_{ij})^2} \tag{3}$$

where p is the number of markers and n_i is the number of alleles for marker i. Let p_{ij} and q_{ij} denote the allele frequencies of allele j for marker i respectively. Note that marker data should be coded -1 and 1 for homozygous genotypes and 0 for heterozygous. If marker data is coded as 0/1/2, data is transformed by function `rogers`, which computes relationship based on Roger’s distance. Using transformation of Hayes and Goddard (2008) rogers distance is related to relationship as

$$f = \frac{s - s_{min}}{1 - s_{min}},$$

with $s = 1 - d$ and s_{min} minimum of all $\frac{n}{2}(n+1)$ values for s. Using rogers distance to compute relationship based on marker data gives

```
> ro <- rogers(M, correction = "Hayes")
> round(ro, 3)

     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  2.0  0.8  0.8  0.4  0.0  0.4
[2,]  0.8  2.0  1.2  0.8  0.4  0.8
[3,]  0.8  1.2  2.0  1.6  1.2  0.8
[4,]  0.4  0.8  1.6  2.0  1.6  0.4
[5,]  0.0  0.4  1.2  1.6  2.0  0.0
[6,]  0.4  0.8  0.8  0.4  0.0  2.0
attr(,"class")
[1] "relationshipMatrix"
```

6.3 Doubled haploid lines

In plant breeding, doubled haploid (DH) lines are used, e.g. in maize. DH lines are fully inbred and thus have an inbreeding coefficient of 1. This has to be taken into account, when the relationship matrix in a pedigree with DH lines is computed. As an example the `maize` data is taken.

```
> data(maize)
> head(maize.ped)
```

```
  ID Par1 Par2 DH
1  1    0    0  1
2  2    0    0  1
3  3    0    0  1
4  4    0    0  1
5  5    0    0  1
6  6    0    0  1
```

First, the additive numerator relationship matrix is computed. There are 1276 DH lines and 25 non DH lines in the pedigree. For DH lines special treatment is necessary, as the inbreeding coefficient must be 1. An argument `DH` is available for function `kinship` where for each individual in the pedigree it specified whether this is a DH line or not. This information is available for the `maize` data. To obtain the additive numerator relationship matrix, use

```
> ped.maize <- create.pedigree(maize.ped$ID, maize.ped$Par1,
+     maize.ped$Par2)
> A.maize <- kinship(ped.maize, DH = maize.ped$DH, ret = "add")
> dim(A.maize)
```

```
[1] 1301 1301
```

6.4 Visualization of relationship matrices

As in most cases a relationship matrix is too big to show completely. Thus there are two possibilities for visualization of an object of class ”relationshipMatrix” in `synbreed` package. A `summary` method is defined which gives the important characteristics of a relationship matrix. Use

```
> summary(A.maize)
```

```
Dimension         : 1301 x 1301
Rank              : 1276
Range             : 0 -- 2
# of unique values: 6
```

to get the summary for the pedigree based additive relationship matrix of the `maize` data. Another possibility is the `plot` method which could be applied to an object of class ”relationshipMatrix”. This gives a heatmap of the entries of the relationship matrix

Note that objects of class ”relationshipMatrix” can be written two input files appropriate for Mixed Model software as `WOMBAT` (Meyer, 2006) or `ASReml` (Gilmour et al., 2000).

7 Models

The ultimate aim in the analysis of a breeding program is to estimate (genomic) breeding values (additive genetic effects). The basic statistical model for this purpose is a linear mixed model

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{Z}\mathbf{u} + \mathbf{e}, \tag{4}$$

where $\mathbf{y}$ is the $n \times 1$ vector of phenotypic records, $\mathbf{b}$ a $t \times 1$ vector of fixed effects and $\mathbf{u}$ a $m \times 1$ vector of random effects. $\mathbf{X}$ and $\mathbf{Z}$ are the corresponding design matrices with dimension $n \times t$ and $n \times m$ respectively. For the random effect, it is assumed that

$$\mathbf{u} \sim \mathrm{N}(\mathbf{0}, \mathbf{G}\sigma^{\mathbf{2}}_{\mathbf{g}})$$

where $\mathbf{G}$ is a variance-covariance matrix. $\mathbf{e}$ denotes the $n \times 1$ vector of residuals with $\mathbf{e} \sim \mathrm{N}(\mathbf{0}, \mathbf{I_n}\sigma^{\mathbf{2}})$ and $\mathbf{I_n}$ is the n-dimensional identity matrix.

If $\mathbf{G}$ equals the additive numerator relationship matrix $\mathbf{A}$, model (4) is called *animal model*. Here the random effect is usually denoted as $\mathbf{a}$ and $\mathbf{a} \sim \mathrm{N}(\mathbf{0}, \mathbf{A}\sigma^{\mathbf{2}}_{\mathbf{a}})$. This model is used to estimate breeding values based on phenotypic records and pedigree information. As an example we will consider simulate plant breeding data. A pedigree with 5 generations and 20 individuals in each generation is simulated. Phenotypic data was measured in a field trial consisting of 5 locations with two replications (blocks) within locations for each of the $n = 100$ genotypes.

The simulation of phenotypes is done with the function `simul.phenotype` which simulates records based on model (4) with an overall mean as fixed effects and random effects for genotype, location and block. Random effects for location, block nested in location and residuals are i.i.d. normal with mean zero and given variance component. Random effects for genotype is simulated according to multivariate normal distribution $\mathrm{N}(\mathbf{0}, \mathbf{A}\sigma^{\mathbf{2}}_{\mathbf{a}})$, where $\mathbf{A}$ is the numerator relationship matrix obtained by pedigree information. The additive genetic variance σ^2_a and variance components for location, block and residual are specified by the user. Simulated data is obtained as follows

```
> ped <- simul.pedigree(5, 20)
> vc <- list(sigma2e = 15, sigma2a = 10, sigma2l = 0, sigma2b = 0)
> dat <- simul.phenotype(ped, Nloc = 5, Nrepl = 2, vc = vc)
> str(dat)
```

```
'data.frame':        1000 obs. of  5 variables:
 $ ID   : Factor w/ 100 levels "1","10","100",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Loc  : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 2 3 3 4 4 5 5 ...
 $ Block: Factor w/ 2 levels "1","2": 1 2 1 2 1 2 1 2 1 2 ...
 $ Trait: num  102.7 101.2 97 98.7 95.7 ...
 $ TBV  : num  -1.55 -1.55 -1.55 -1.55 -1.55 ...
```

Variance components for location and block are zero as only additive genetic effects should be considered in this example. The simulated random effects for genotype are called true breeding values (TBV).

Estimation of variance components with REML and prediction of random effects in model (4) could be done with function `regress` in package `regress`. This package allows for arbitrary variance-covariance matrices of random effects. Solutions for animal model with overall mean as fixed effect are obtained as

```
> library(regress)
> A <- kinship(ped, ret = "add")
> A <- A %x% matrix(1, 10, 10)
> mod <- regress(Trait ~ 1, Vformula = ~A, data = dat)
> summary(mod)


Maximised Residual Log Likelihood is -1950.051

Linear Coefficients:
             Estimate Std. Error
 (Intercept)   99.897      0.762

Variance Coefficients:
             Estimate Std. Error
          A    10.349      1.938
          In   15.544      0.731
```

Note that variance-covariance matrix must be of same dimension as y. This could easily be obtained by using the Kronecker product to enlarge relationship matrix as data is sorted by individuals. As the fitted model only contains one random effect, estimated breeding values are

```
> ebv <- mod$predicted - mod$fitted
```

as `predicted` equals $\hat{\mathbf{y}}$ and `fitted` contains estimated overall mean. Correlation between observed and estimated phenotypes (also called *predictive ability* of the model, see Legarra et al. (2008)) is

```
> cor(ebv, dat$Trait)


          [,1]
[1,] 0.5823288
```

and correlation between estimated and true genetic effect (called *accuracy* of the model) is

```
> cor(ebv, dat$TBV)
```

```
          [,1]
[1,] 0.9067806
```

At the end of one cycle of a breeding scheme the individuals with highest breeding values are selected for the next breeding cycle. The term *genomic selection* refers to the situation where genotypic data is used to estimate genetic effects of the individuals. This is the case if a genomic relationship matrix is used as variance-covariance matrix in the animal model. In `maize` data genotypic data is available for all individuals with phenotypes. This is used to set up genomic relationship according to vanRaden and passed to `regress`

```
> data(maize)
> marker <- codeGeno(maize.geno[, -1])
> vR <- vanRaden(marker)
> y <- maize.pheno$Trait
> mod.maize <- regress(y ~ 1, Vformula = ~vR)
> summary(mod.maize)
```

```
Maximised Residual Log Likelihood is -3028.19

Linear Coefficients:
             Estimate Std. Error
 (Intercept) 1194.173      0.172

Variance Coefficients:
             Estimate Std. Error
          vR   16.681      2.485
          In   37.145      1.686
```

Note that we have only one phenotypic record for each genotype Predictive ability of this model is

```
> yhat <- mod.maize$predicted
> cor(yhat, y)
```

```
          [,1]
[1,] 0.6925263
```

Genomic selection is done by choosing the, e.g. 20% genotypes with highest values for $\hat{\mathbf{y}}$.

8 Acknowledgements

This research was funded by the German Federal Ministry of Education and Research (BMBF) within the AgroClustEr *Synbreed – Synergistic plant and animal breeding.*

References

Bernardo, R. (2002). *Breeding for Quantitative Traits in Plants.* Stemma Press.

Gilmour, A., BR, C., SJ, W., and R, T. (2000). ASREML. program user manual. *NSW Agriculture, Orange Agricultural Institute, Forest Road, Orange, Australia.*

Hayes, B. J. and Goddard, M. E. (2008). Technical note: Prediction of breeding values using marker-derived relationship matrices. *J. Anim Sci.*, 86(9):2089–2092.

Legarra, A., Robert-Granie, C., Manfredi, E., and Elsen, J. (2008). Performance of genomic selection in mice. *Genetics*, 180(1):611–618.

Meuwissen, T. H. E., Hayes, B. J., and Goddard, M. E. (2001). Prediction of total genetic value using Genome-Wide dense marker maps. *Genetics*, 157(4):1819–1829.

Meyer, K. (2006). Wombat - a tool for mixed model analyses in quantitative genetics by reml. *J. Zhejinag Uni SCIENCE*, pages 815–821.

Oakey, H., Verbyla, A. P., Cullis, B. R., Wei, X., and Pitchford, W. S. (2007). Joint modelling of additive and non-additive (genetic line) effects in multi-environment trials. *Theoretical and Applied Genetics*, (114):1319–1332.

Rogers, J. (1972). Measures of genetic similarity and genetic distance. In *Studies in genetics VII*, volume 7213, pages 145–âĂŞ153. Univ. of Texas, Austin.

vanRaden, P. (2008). Efficient methods to compute genomic predictions. *Journal of Dairy Science*, 91:4414–4423.

Wright, S. (1922). Coefficients of inbreeding and relationship. *Am Nat*, pages 330–338.