
AGHmatrix Tutorial

R package to compute relationship matrices for diploid and
autotetraploid species

Rodrigo R Amadeu¹, Catherine Cellon², Márcio F R Resende Jr.², James W
Olmstead², A Augusto F Garcia¹, and Patricio R Munoz²

Department of Genetics¹
“Luiz de Queiroz” College of Agriculture (ESALQ)
University of São Paulo (USP) - Brazil

Institute of Food and Agricultural Sciences (IFAS)²
University of Florida (UF) - USA

This package and its tutorial are maintained by Rodrigo Amadeu and Patricio Muñoz. If you find errors or have suggestions please send them to rramadeu@gmail.com.

Contents

Contents	2
1 Overview	2
1.1 Citation	3
1.2 About R	3
1.3 Installing the package	3
1.4 Loading AGHmatrix package	4
2 Building relationship matrices	4
2.1 Relationship matrices using the pedigree data - A matrix	4
2.2 Relationship matrices with molecular data - G Matrix	5
2.3 Covariance matrices due to epistatic terms	6
3 Exporting your data to be used in ASReml - csv format	7
3.1 Making a loop in order to get several matrices	8
Bibliography	10

1 Overview

AGHmatrix software is an R-package under development which builds relationship matrices based on pedigree (A matrix) and/or molecular markers (G matrix), and in the future with the possibility to build a combined matrix of pedigree corrected by molecular information (H matrix). The package also works for diploid and autotetraploid data and was firstly developed for the study of [Amadeu *et al.* \(2016\)](#).

To build the A-matrix for diploids, it uses the method proposed by [Henderson \(1976\)](#), described in [Mrode \(2014\)](#). The package can build additive and dominance relationship matrices for diploid species.

To build the A-matrix for autotetraploid, it uses the method proposed by [Kerr *et al.* \(2012\)](#) and described in [Slater *et al.* \(2014\)](#). The package builds additive relationship matrices only for tetraploids.

To build the G-matrix for diploids, the user can chose from two methods for the additive relationship matrix: as described in [Yang *et al.* \(2010\)](#) or as described in [VanRaden \(2008\)](#). The user can also chose to build dominance relationship matrices based either on [Su *et al.* \(2012\)](#) or on [Vitezica *et al.* \(2013\)](#).

The G-matrix for autotetraploids is is not implemented yet.

The user can build covariance matrices due to epistasis using the Hadamard products (as in Muñoz *et al.* (2014)) for additive, or dominance matrices, coming from pedigree or molecular information.

1.1 Citation

This software should be cited as:

Amadeu, R.R., Cellon, C., Olmstead, J.W., Garcia, A.A.F., Resende, M.F.R., and Muñoz, P.R. "AGHmatrix: R Package to Construct Relationship Matrices for Autotetraploid and Diploid Species: A Blueberry Example." The Plant Genome (2016), Vol 9, No 3.

1.2 About R

R (R Core Team 2016) is a free programming language widely used in statistical computing. To download R, please visit the Comprehensive R Archive Network (<http://cran.r-project.org>). Users can also install the software Rstudio, which present a more intuitive way to use R. To download Rstudio please go to (<https://www.rstudio.com/products/RStudio/>).

For a quick start, we recommend to follow:

- Our Introduction to R tutorial available at <http://augusto-garcia.github.io/statgen-esalq/Introduction-to-R/>.
- Our R Introductory presentation available at <http://augusto-garcia.github.io/R-Introduction/>.
- The "Introduction to R" section in "OneMap Tutorial" available at <http://cran.r-project.org/web/packages/onemap/index.html> for a quick introduction.
- And the "Verzani's simpleR — Using R for Introductory Statistics" available at <http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf> for a deeper introduction.

1.3 Installing the package

After you have R installed in your machine, you can install the AGHmatrix package. Within R, you need to install and load the package devtools, to automatically build and install packages from the github platform:

```
install.packages("devtools")
library(devtools)
```

If you use Windows, first install package Rtools . On a Mac, you will need Xcode (available on the App Store). On Linux, you are ready to go.

Then, to install AGHmatrix from github use this:

```
install_github("prmunoz/AGHmatrix")
```

1.4 Loading AGHmatrix package

To loading the package:

```
library(AGHmatrix)
```

The package should be available in your R package active list.

2 Building relationship matrices

2.1 Relationship matrices using the pedigree data - A matrix

In this section we presented how to load the data into the software and how to construct the pedigree-based relationship matrix (A-matrix) for diploid and autotetraploid species. In the package, the function `Amatrix` handles the pedigree and build the A-matrix related to that given pedigree. The matrix is build according to the recursive method presented in [Mrode \(2014\)](#) and described by [Henderson \(1976\)](#). This method is expanded for higher ploidy (n-ploidy) according with [Kerr *et al.* \(2012\)](#) described in [Slater *et al.* \(2014\)](#).

After loading the package you have to load your data file into the software. To do this, you can use the function `read.data()` or `read.csv()` (If specifically the format `.csv` file is used) for example. Your data should be available in R as a dataframe structure in the following order: column 1 must be the individual/genotype names (id), columns 2 and 3 must be the parent names. In the package there is a pedigree data example (`ped.mrode`) that can be used to look at the structure and order the data. To call `ped.mrode`:

```
data(ped.mrode)
ped.mrode
class(ped.mrode)
```

The example `ped.mrode` has 3 columns, column 1 contains the names of the individual/genotypes, column 2 contains the names of the first parent, column 3 contains the names of the second parental. There is no header and the unknown value default is 0. Your data has to be in the same format of `ped.mrode`.

In the algorithm, the first step is the pre- processing of the pedigree: the individuals are numerated 1 to n . Then, it is verified whether the genotypes in the pedigree are in chronological order (*i.e.* if the parents of a given individual are located prior to this individual in the pedigree dataset). If this order is not followed, the algorithm performs the necessary permutations to correct them. After this pre-processing, the matrices computation proceeds as in [Mrode \(2014\)](#) for diploid - for additive or dominance relationship - and as in [Slater *et al.* \(2014\)](#) for autotetraploids - for additive relationship.

To build the relationship matrix you need to type the function `Amatrix` with the following arguments: `data`, `ploidy`, `double reduction`, `unknown values`. If you want a dominance relationship matrix you also need to use the argument `dominance` as showed in the next chunk.

For example, if ploidy is equal to 2 and unknown values are identify as 0 the matrix will be calculated as presented in [Mrode \(2014\)](#), Chapter 2, with the following code:

```
# For additive relationship matrix
Amatrix(data=ped.mrode,ploidy=2,unk=0)
# For dominance relationship matrix
Amatrix(data=ped.mrode,ploidy=2,unk=0,dominance=TRUE)
```

If ploidy is equal to 4 and double reduction equals to 10%, the matrix is calculated as presented in [Slater *et al.* \(2014\)](#) with the following code:

```
# For additive relationship matrix
Amatrix(data=ped.mrode,ploidy=4,w=0.1,unk=0)
```

If you want to save your matrix in an object, you can use the following code:

```
MyMatrix <- Amatrix(data=ped.mrode,ploidy=4,w=0.1,unk=0)
```

To obtain more information about the `Amatrix` function you can type:

```
?Amatrix
```

2.2 Relationship matrices with molecular data - G Matrix

This section presents how to load the data and how to construct the genomic-based relationship matrix for diploid species. In the package, the function `Gmatrix` is the one that handles the molecular-markers matrix and builds the relationship matrix. Molecular markers data should be organized in a matrix format (individual in rows and markers in columns) coded as 0,1,2 and missing data value (numeric or NA). Import your molecular marker data into R with the function `read.table()` and convert to a matrix format with the function `as.matrix()`. The function `Gmatrix` can be used then to construct the additive relationship either as proposed by [Yang *et al.* \(2010\)](#) or the proposed by [VanRaden \(2008\)](#). The function can also construct the dominance relationship matrix either as proposed by [Su *et al.* \(2012\)](#) or as proposed by [Vitezica *et al.* \(2013\)](#).

As an example, here we build the four matrices using real data from [Resende *et al.* \(2012\)](#). The data (`snp.pine`, which is part of this R package, contains the marker data from [Resende *et al.* \(2012\)](#).

```

#loading the marker data example
data(snp.pine)

#verifying the data structure, must be matrix
class(snp.pine)

#looking the first 3x3 elements of the matrix
#snp.table missing values is coded as -9.
#individuals on rows and markers on columns.
snp.pine[1:3,1:3]

#building the additive relationship matrix based on VanRaden 2008
G.VanRaden <- Gmatrix(SNPmatrix=snp.pine,
                      missingValue=-9, maf=0.05, method="VanRaden")

#building the additive relationship matrix based on Yang 2010
G.Yang <- Gmatrix(SNPmatrix=snp.pine,
                  missingValue=-9, maf=0.05, method="Yang")

#building the dominance relationship matrix based on Su 2012
G.Yang <- Gmatrix(SNPmatrix=snp.pine,
                  missingValue=-9, maf=0.05, method="Su")

#building the dominance relationship matrix based on Vitezica 2013
G.Yang <- Gmatrix(SNPmatrix=snp.pine,
                  missingValue=-9, maf=0.05, method="Vitezica")

```

More information about the Gmatrix function can be found by typing:

```
?Gmatrix
```

2.3 Covariance matrices due to epistatic terms

Here we present how to easily compute the epistasis relationship matrices using Hadamard products (*i.e.* cell-by-cell product), denoted by #. For more information please see [Muñoz *et al.* \(2014\)](#). In this example we are using the molecular-based relationship matrix.

First, build the additive and dominance matrices:

```

A<- Gmatrix(SNPmatrix=snp.pine,
             method="VanRaden",missingValue=-9,maf=0.05)
D <- Gmatrix(SNPmatrix=snp.pine,
             ,method="Vitezica",missingValue=-9,maf=0.05)

```

For the first degree epistatic terms:

```
#Additive-by-Additive Interactions
A_A <- A*A
#Dominance-by-Additive Interactions
D_A <- D*A
#Dominance-by-Dominance Interactions
D_D <- D*D
```

For the seconde degree epistatic terms:

```
#Additive-by-Additive-by-Additive Interactions
A_A_A <- A*A*A
#Additive-by-Additive-by-Dominance Interactions
A_A_D <- A*A*D
#Additive-by-Dominance-by-Dominance Interactions
A_D_D <- A*D*D
#Dominance-by-Dominance-by-Dominance Interactions
D_D_D <- D*D*D
```

3 Exporting your data to be used in ASReml - csv format

In this section, we present how to use the function `formatmatrix` to export a recently build matrix in the format compatible with ASReml standalone version. That is the lower diagonal matrix formatted in three columns in .csv format (other ASCII extension could be used as well). In order to do this, we need to build a matrix, its inverse, and export it using `formatmatrix` function. ASReml can invert the relationship matrix as well, probably more efficiently than R for large matrices (*i.e.* `solve()` function), so no need to invert the matrix in R if matrix is large. This function has as options: `round.by`, which let you decide the number of decimals you want; `exclude.0`, if TRUE, remove all the zeros from your data; and, `name` that defines the name to be used in the exported file. Use the default if not sure what parameter use in these function.

Here an example using the mrode pedigree data:

```
#setting the number of digits to display in R for 12
options(digits=12)

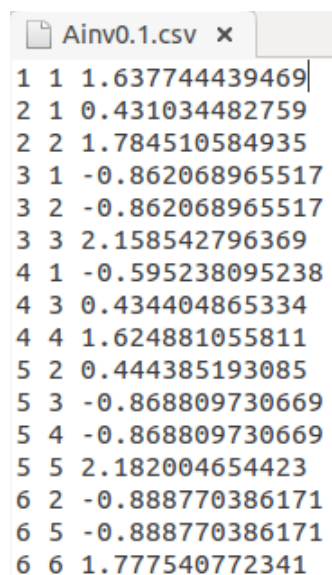
#loading the data example
data(ped.mrode)

#building the matrix
A<-Amatrix(data=ped.mrode, ploidy=4, w=0.1, unk=0)
```

```
#build the inverse
Ainv<-solve(A)

#exporting it. The function "formatmatrix" will convert
#the matrix in a 3-column table.
formatmatrix(Ainv, round.by=12, exclude.0=TRUE, name="Ainv0.1")
```

This script will create the following csv file presented in Figure 1.



Row ID	Column ID	Inverse Value
1	1	1.637744439469
2	1	0.431034482759
2	2	1.784510584935
3	1	-0.862068965517
3	2	-0.862068965517
3	3	2.158542796369
4	1	-0.595238095238
4	3	0.434404865334
4	4	1.624881055811
5	2	0.444385193085
5	3	-0.868809730669
5	4	-0.868809730669
5	5	2.182004654423
6	2	-0.888770386171
6	5	-0.888770386171
6	6	1.777540772341

Figure 1: Inversed A-matrix from ped.mrode example. Data is arranged in a lower diagonal sorter in a three column format csv file that ASReml will accept directly. The first 2 columns represent the rows and columns IDs of the matrix while the third column contains the inverse relationship value. All the rows with value equal to 0 were excluded from the file. Note that diagonal elements should be present for ASReml to work.

3.1 Making a loop in order to get several matrices

In this section, we present a simple for function for the user to be able to obtain in a practical way several matrices for different double reduction values (if polyploidy) to later be used in ASReml (for example).

```
#setting the number of digits to display in R for 12
options(digits=12)
```



```

#loading the data example
data(ped.mrode)

#determining your double reduction range
double.red<-seq(0,0.2,0.05)

#extracting the length of double.red
n<-length(double.red)

#making the loop
for(i in 1:n){
  A<-Amatrix(data=ped.mrode,
             ploidy=4,
             w=double.red[i],
             unk=0)
  #making the inverse
  A.inv<-solve(A)
  #exporting as csv
  formatmatrix(data=A.inv,
               name=paste("Ainv_",double.red[i],sep=""),
               round.by=12,
               exclude.0=TRUE)
}

```

At the end, this script will write 5 files representing 5 matrices (with different levels of double-reduction proportion specified; 0, 0.05, 0.1, 0.15, and 0.2). These matrices will be in a 3 column-way format as in Figure 1.

Bibliography

- Amadeu, R. R., C. Cellon, J. W. Olmstead, A. A. Garcia, M. F. Resende, and P. R. Muñoz, 2016 AGHmatrix: R package to construct relationship matrices for autotetraploid and diploid species: a blueberry example. *The Plant Genome* **9**.
- Henderson, C., 1976 A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. *Biometrics* pp. 69–83.
- Kerr, R. J., L. Li, B. Tier, G. W. Dutkowski, and T. A. McRae, 2012 Use of the numerator relationship matrix in genetic analysis of autopolyploid species. *Theoretical and Applied Genetics* **124**: 1271–1282.
- Mrode, R. A., 2014 *Linear models for the prediction of animal breeding values*. Cabi.
- Muñoz, P. R., M. F. Resende, S. A. Gezan, M. D. V. Resende, G. de los Campos, M. Kirst, D. Huber, and G. F. Peter, 2014 Unraveling additive from nonadditive effects using genomic relationship matrices. *Genetics* **198**: 1759–1768.
- R Core Team, 2016 *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Resende, M. F., P. Muñoz, M. D. Resende, D. J. Garrick, R. L. Fernando, J. M. Davis, E. J. Jokela, T. A. Martin, G. F. Peter, and M. Kirst, 2012 Accuracy of genomic selection methods in a standard data set of loblolly pine (*pinus taeda* l.). *Genetics* **190**: 1503–1510.
- Slater, A. T., G. M. Wilson, N. O. Cogan, J. W. Forster, and B. J. Hayes, 2014 Improving the analysis of low heritability complex traits for enhanced genetic gain in potato. *Theoretical and applied genetics* **127**: 809–820.
- Su, G., O. F. Christensen, T. Ostensen, M. Henryon, and M. S. Lund, 2012 Estimating additive and non-additive genetic variances and predicting genetic merits using genome-wide dense single nucleotide polymorphism markers. *PloS one* **7**: e45293.
- VanRaden, P., 2008 Efficient methods to compute genomic predictions. *Journal of dairy science* **91**: 4414–4423.
- Vitezica, Z. G., L. Varona, and A. Legarra, 2013 On the additive and dominant variance and covariance of individuals within the genomic selection scope. *Genetics* **195**: 1223–1230.
- Yang, J., B. Benyamin, B. P. McEvoy, S. Gordon, A. K. Henders, D. R. Nyholt, P. A. Madden, A. C. Heath, N. G. Martin, G. W. Montgomery, *et al.*, 2010 Common snps explain a large proportion of the heritability for human height. *Nature genetics* **42**: 565–569.