Avenida Pádua Dias, 11. Caixa Postal 83
Piracicaba, São Paulo, Brazil. 13400-970
Phone: +55 19 3429-4125
rodrigo.amadeu@usp.br

AGHmatrix Tutorial

R package to compute and analyze relationship matrices for diploid and autotetraploid species

Rodrigo R Amadeu[1], Catherine Cellon[2], Márcio F R Resende Jr.[2], James W. Olmstead[2], A Augusto F Garcia[1], and Patricio R Munoz[2]

Department of Genetics[1]
"Luiz de Queiroz" College of Agriculture (ESALQ)
University of Sao Paulo (USP) - Brazil

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

Piracicaba, São Paulo
Brazil
March 31, 2016

Contents

1 Overview

AGHmatrix software is an R-package under development mainly to build relationship matrices using pedigree (A matrix) and/or molecular markers (G matrix) with the possibility to build a combined matrix of Pedigree corrected by Molecular (H matrix). The package also works with Diploid and Autotetraploid Data.

For the pedigree diploid data, it uses the method proposed by Henderson (1976) and described in Mrode (2014).

For the pedigree autotetraploid data, it uses the method proposed Kerr *et al.* (2012) and described in Slater *et al.* (2014).

For the molecular diploid data, it can use 2 methods: Powell *et al.* (2010) and VanRaden (2008).

For the molecular autotetraploid data, it uses a variation of diploid methods which are under development.

The combined matrix H is under development.

1.1 Citation

How to cite this software:

Available soon...

2 About R

R (R Core Team 2012) 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). An alternative is to install the RStudio software, it is a more intuitive/graphical way to use R. To download it, please go to (https://www.rstudio.com/products/RStudio/).

For I quick start, we recommend to follow:

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

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:

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

This will allow you to automatically build and install packages from github. If you use Windows, first install Rtools package Rtools . On a Mac, you will need Xcode (available on the App Store). On Linux, you are good to go.

Then, to install AGHmatrix from github:

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

3.1 Loading AGHmatrix package

After, open R (or RStudio) and type:

```
library(AGHmatrix)
```

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

4 Loading your pedigree file

After load the package you have to load your data file. For it, you can use the function `read.data()` or `read.csv()` for example.
Your data should be available in R as a dataframe where column 1 should be the individual names (id), column 2 and 3 should be the parent names. In the package there is a data example. To look it, type:

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

##    Ind Par1 Par2
## 1 Anc1    0    0
## 2 Anc2    0    0
## 3 Var1 Anc1 Anc2
## 4 Var2 Anc1    0
## 5 Var3 Var2 Var1
## 6 Var4 Var3 Anc2
```

The example *ped.mrode* (above) has 3 columns, where column 1 is the individual names, column 2 is the parental 1 names, column 3 is the parental 2 names. There is no header and the unknown value default is 0. Your personal data has to be in the same format than *ped.mrode*.

5 Relationship Matrix with Pedigree Data

In this section is presented how to load the data and how to construct the relationship matrix for diploid and autotetraploid species. The pedigree-base relationship matrix calculation, matrix A, is performed according with a recursive method as presented in Mrode (2005) and described by Henderson (1976). This method is expanded for n-ploidy according with Kerr *et al.* (2012) described in Slater *et al.* (2014).
In the algorithm, first occurs the preprocessing of the data. To the preprocessing of the pedigree, first, the individuals are numerated 1 to , where is the total individuals of the pedigree data. Then, it is verified if they are chronological sorted (i.e., if the parents of a given individual n are located before it in the list). If not, the algorithm performs necessary permutes. After preprocessing, occurs the matrix computation as presented in Mrode (2014) for diploid and Slater *et al.* (2014) for autotetraploidy.

5.1 Building matrix A

To build the A matrix you need to type the function with the data, ploidy, double reduction and unknown values. For example, if ploidy equals to 2 and unknown value equals 0 is calculated as presented in Mrode (2014) with the following code:

```
Amatrix(data=ped.mrode,ploidy=2,unk=0)
```

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

```
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:

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

More information about the Amatrix function you can have typing:

```
?Amatrix
```

5.2 Exporting your data as ASREML csv format

In this section, we present how to use the function formatmatrix in order to export a matrix to a compatible ASREML format (csv file with 3 columns). In order to do it, we need to build a matrix, its inverse, and export it using *formatmatrix* function. This function has as options: *round.by*, which set the number of decimals you desite, *exclude.0*, if TRUE, remove all the zeros from your data, *name* what is the desired name of your file.
Below, an example of how to do it:

```
#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 2.

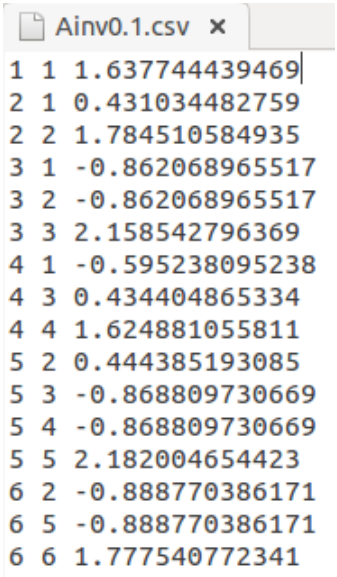

Figure 1: csv file representing an inversed A matrix from ped.mrode data with w=0.1. The first 2 columns represent rows and columns of the matrix, the third column represents the value. All the rows with value equal to 0 it was excluded from the file.

5.3 Making a *loop* in order to get several matrices

In this section, we present a simple "for" function for the user be able to get in a pratical way several matrices for different double reduction values to later be used in ASREML (for example). In R:

```
#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)

#Looping it
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, it will get 5 files represents 5 matrices (if double-reduction proportion of 0, 0.05, 0.1, 0.15, and 0.2). These matrices will be in a 3 column-way format as in Figure 2.

6 Relationship Matrices with Molecular Data - G Matrix

This chapter is under construction!

To build the relationship matrix based on markers, your data should be organized in a matrix (markers x individual) coded as 0,1,2 and missing data value. Your data can be easily loaded in R with the function *read.table()*. The function *Gmatrix* can construct the matrix proposed by VanRaden (2008) or the matrix proposed by Powell *et al.* (2010).

As example, here we build both matrices using fake data (*snp.table*, which is part of this R package).

```
#loading the data example
data(snp.table)

#looking the data, snp.table missing values is coded as -9
snp.table

##          Genotype2 Genotype3 Genotype8
## Marker1          2         2         2
## Marker2          2         2         2
## Marker3          2         2         2
## Marker4          2         2         2
## Marker5          1        -9         2
## Marker6         -9         2        -9
## Marker7          2         2         2
## Marker8          2         1         2
## Marker9          2         2         2
## Marker10         2         2         1
```

```
## Marker11         0         1         1
## Marker12         2         2         2
## Marker13         1         0         1
## Marker14         2         2         2
## Marker15         1         1         2
## Marker16         2         1         1
## Marker17         2         2         2
## Marker18         1         2         2
## Marker19         2         0         2
## Marker20         2         2         2
## Marker21         1         2         1
## Marker22         2         2         2
## Marker23         2         2         2
## Marker24         2        -9         1
## Marker25         0         2         2
## Marker26         0         1         0
## Marker27         1         1         1
## Marker28         1         2         2
## Marker29         2         0         0
## Marker30         2         2         2

#building the matrix based upon VanRaden
G<-Gmatrix(SNPdata=snp.table, missingValue=-9, method="VanRaden")

##  Number of Markers: 30
##  Number of Individuals: 3
##
## Completed! Time = 0.002  seconds

G

##             Genotype2  Genotype3  Genotype8
## Genotype2  1.55156951 -0.2511211 0.09865471
## Genotype3 -0.25112108  1.2556054 0.39461883
## Genotype8  0.09865471  0.3946188 1.22869955

#building the matrix based upon Powell
G<-Gmatrix(SNPdata=snp.table, missingValue=-9, method="Powell")

##  Number of Markers: 30
##  Number of Individuals: 3
##
## Completed! Time = 0  seconds

G
```

```
##             Genotype2   Genotype3 Genotype8
## Genotype2  1.14111111 -0.09166667 0.1388889
## Genotype3 -0.09166667  1.05333333 0.2233333
## Genotype8  0.13888889  0.22333333 1.0811111
```

More information about the Gmatrix function you can have typing:

```
?Gmatrix
```

To invert and export the matrix follow the steps already described in 5.2.

Bibliography

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.

Powell, J. E., P. M. Visscher, and M. E. Goddard, 2010 Reconciling the analysis of IBD and IBS in complex trait studies. Nature reviews. Genetics **11**: 800–805.

R Core Team, 2012 *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

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.

VanRaden, P., 2008 Efficient methods to compute genomic predictions. Journal of dairy science **91**: 4414–4423.