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 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
October 9, 2016

AGHmatrix package version 0.18. Updated at 09/oct/2016

This package and tutorial is maintained by Rodrigo Amadeu and Patricio Muñoz. If you find any errors or suggestions please send it to rramadeu@gmail.com.

Contents

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 (H matrix). The package also works with Diploid and Autotetraploid Data and was firstly developed for the study of Amadeu *et al.* (2016).

For the pedigree diploid data, it uses the method proposed by Henderson (1976), described in Mrode (2014), and it can build additive and dominance relationship matrices.

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

For the molecular diploid data, it can uses two methods for additive relationship matrices - described in Yang *et al.* (2010) or VanRaden (2008) - and two methods for dominance relationship matrices - described in Su *et al.* (2012) or Vitezica *et al.* (2013).

For the molecular autotetraploid data, it is not implemented/developed yet.

The combined matrix H is not implemented/developed yet.

The user can build covariance matrices due to epistatis using Hadamard products (as in Muñoz *et al.* (2014))

1.1 Citation

How to cite this software:
Amadeu, Rodrigo R., Catherine Cellon, James W. Olmstead, Antonio AF Garcia, Marcio FR Resende, and Patricio R. Muñoz. ”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). An alternative is to install the RStudio software which is a more intuitive 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 Introduction to R tutorial available at http://augusto-garcia.github.io/statgen-esalq/Introduction-to-R/.

- 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.

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:

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

This will allow you to automatically build and install packages from github platform. 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”)
```

1.4 Loading AGHmatrix package

To load the package type:

```
library(AGHmatrix)
```

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

2 Building relationship matrices

2.1 Relationship matrices with pedigree data - A matrix

In this section is presented how to load the data and how to construct the pedigree-based relationship matrix for diploid and autotetraploid species. In the package, the function `Amatrix` is the one which handle pedigree and build its relationship matrix. The matrix 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 the necessary permutes. After preprocessing, occurs the matrix computation as presented in Mrode (2014) for diploid - for additive or dominance relationship - and Slater *et al.* (2014) for autotetraploidy - for additive relationship.

After load the package you have to load your data file. To do 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 pedigree data example. To look it, type:

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

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 data has to be in the same format of `ped.mrode`. To build the relationship matrix you need to type the function with the following arguments: data, ploidy, double reduction, unknown values and, if you want dominance relationship. For example, if ploidy equals to 2 and unknown value equals 0 is calculated as presented in Mrode (2014), Chapter2, 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 equals to 4 and double reduction equals to 10% 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)
```

More information about the Amatrix function you can have typing:

```
?Amatrix
```

2.2 Relationship matrices with molecular data - G Matrix

In this section is presented 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 which handle molecular-markers matrix and build its relationship matrix. To build the relationship matrix based on markers, your data should be organized in a matrix format (individual x marks) coded as 0,1,2 and missing data value. Your data can be easily loaded in R with the function `read.table()` and converted with `as.matrix()` function. The function `Gmatrix` can construct the additive relationship matrix proposed by Yang *et al.* (2010) or the one proposed by VanRaden (2008). The function can also construct the dominance relationship matrix proposed by Su *et al.* (2012) or Vitezica *et al.* (2013). As an example, here we build the four matrices using real data from Resende *et al.* (2012) (`snp.pine`, which is part of this R package).

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

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

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

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

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

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

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

More information about the Gmatrix function you can have typing:

```
?Gmatrix
```

2.3 Covariance matrices due to epistatic terms

Here we present how to easily compute the epistatic relationship matrices using Hadamard products (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:

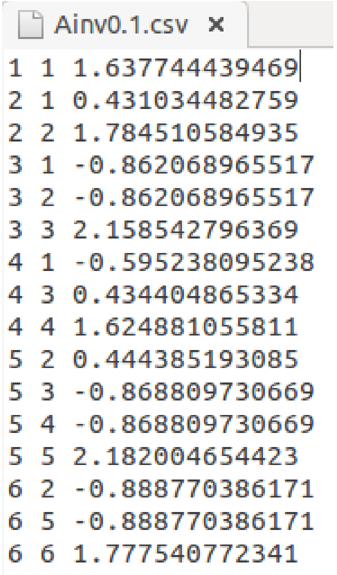
```
#Additive-by-Additive-by-Additive Interactions
A_A_A <- A*A*A
#Additive-by-Additive-by-Dominance Interactions
A_A_D <- A*A*D
#Additive-by-Dominance-by-Dominance Interactions
A_D_D <- A*D*D
#Dominance-by-Dominance-by-Dominance Interactions
D_D_D <- D*D*D
```

3 Exporting your data 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 - standalone - 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 sets the number of decimals you want, `exclude.0`, if TRUE, remove all the zeros from your data, name what will be the name of your file. Use the default if in doubt.
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.



```
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: 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.

3.1 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 writes 5 files represents 5 matrices (with 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.

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. Ostersen, 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.