

# Package ‘AGHmatrix’

September 14, 2021

**Title** Relationship Matrices for Diploid and Autopolyploid Species

**Version** 2.0.4

**Description** Computation of A (pedigree), G (genomic-base), and H (A corrected by G) relationship matrices for diploid and autopolyploid species. Several methods are implemented considering additive and non-additive models.

**Author** Rodrigo Amadeu [aut, cre],  
Catherine Cellon [ctb],  
Leticia Lara [ctb],  
Marcio Resende [ctb],  
Ivone Oliveira [ctb],  
Luis Ferrao [ctb],  
Patricio Munoz [ctb],  
Augusto Garcia [ctb]

**Depends** R (>= 3.6.0)

**Imports** Matrix (>= 1.2-7.1), zoo (>= 1.8.6)

**Suggests** knitr, MASS, rmarkdown

**VignetteBuilder** knitr

**Encoding** UTF-8

**License** GPL-3

**URL** <https://github.com/rramadeu/AGHmatrix>

**Maintainer** Rodrigo Amadeu <rramadeu@ufl.edu>

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-09-14 05:30:02 UTC

## R topics documented:

Amatrix . . . . .	2
AmatrixPolyCross . . . . .	4

datatreat . . . . .	5
filterpedigree . . . . .	6
formatmatrix . . . . .	7
Gmatrix . . . . .	8
Hmatrix . . . . .	10
missingdata . . . . .	12
ped.mrode . . . . .	13
ped.sol . . . . .	14
snp.pine . . . . .	14
snp.sol . . . . .	15

<b>Index</b>	<b>16</b>
--------------	-----------

---

Amatrix	<i>Construction of Relationship Matrix A</i>
---------	--

---

### Description

Creates an additive relationship matrix A from a pedigree data in a 3-column way format based on ploidy level (an even number) and, if ploidy equals 4, based on proportion of parental gametes that are IBD (Identical by Descent) due to double reduction. Returns a dominance relationship matrix if dominance true (ploidy 2 only). Autopolyploid matrices based on Kerr (2012), used when ‘ploidy’ argument is higher than ‘2’ and ‘dominance=FALSE’. Diploid additive numerator relationship matrix built as in Henderson (1976), used when ‘ploidy=2’ and ‘dominance=FALSE’. Diploid dominance numerator relationship matrix built as in Cockerham (1954), used when ‘ploidy=2’ and ‘dominance=FALSE’. For details of recursive method see Mrode (2005).

### Usage

```
Amatrix(
  data = NULL,
  ploidy = 2,
  w = 0,
  verify = TRUE,
  dominance = FALSE,
  slater = FALSE,
  ...
)
```

### Arguments

data	pedigree data name (3-column way format). Unknown value should be equal 0.
ploidy	an even number (default=2).
w	proportion of parental gametas IBD due to double reduction (default=0), only if ploidy=4.
verify	verifies pedigree file for conflictuos entries (default=TRUE).
dominance	if true, returns the dominance relationship matrix

slater            if true, returns the additive autotetraploid relationship matrix as Slater (2013)  
 ...                arguments to be passed to datatreat()

**Value**

Matrix with the Relationship between the individuals.

**Author(s)**

Rodrigo R Amadeu, <rramadeu@gmail.com>

**References**

- Cockerham, CC. 1954. An extension of the concept of partitioning hereditary variance for analysis of covariances among relatives when epistasis is present. Genetics 39, 859–882*
- Henderson, CR. 1976. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32, 69-83*
- Kerr, RJ, et al. 2012. Use of the numerator relationship matrix in genetic analysis of autopolyploid species. Theoretical and Applied Genetics 124 1271-1282*
- Mrode, RA. 2014. Chapter 2: Genetic Covariance Between Relatives and Chapter 9: Non-additive Animal Models in Mrode, RA. 2014. Linear models for the prediction of animal breeding values. Cabi, 3rd edition.*
- Slater, AT, et al. 2013. Improving the analysis of low heritability complex traits for enhanced genetic gain in potato. Theoretical and Applied Genetics 127, 809-820*

**Examples**

```
data(ped.mrode)
#Computing additive relationship matrix considering diploidy (Henderson 1976):
Amatrix(ped.mrode, ploidy=2)
#Computing non-additive relationship matrix considering diploidy (Cockerham 1954):
Amatrix(ped.mrode, ploidy=2, dominance=TRUE)
#Computing additive relationship matrix considering autotetraploidy (Kerr 2012):
Amatrix(ped.mrode, ploidy=4)
#Computing additive relationship matrix considering autooctaploidy (Kerr 2012):
Amatrix(ped.mrode, ploidy=8)
#Computing additive relationship matrix considering autotetraploidy and double-
#reduction of 0.1 (Kerr 2012):
Amatrix(ped.mrode, ploidy=4, w=0.1)
#Computing additive relationship matrix considering
#autotetraploidy and double-reduction of 0.1 (Slater 2014):
Amatrix(ped.mrode, ploidy=4, w=0.1, slater = TRUE)
#Computing additive relationship matrix considering autohexaploidy and double-
#reduction of 0.1 (Kerr 2012):
Amatrix(ped.mrode, ploidy=6, w=0.1)
```

---

AmatrixPolyCross	<i>Construction of pedigree-based relationship matrix with parental guessing possibility</i>
------------------	--

---

### Description

Creates an additive relationship matrix *A* based on a non-deterministic pedigree with 4+ columns where each column represents a possible parent. This function was built with the following designs in mind. 1) A mating design where you have equally possible parents. For example, a generation of insects derived from the mating of three insects in a cage. All the insects in this generation will have the same expected relatedness with all the possible parents (1/3). If there are only two parents in the cage, the function assumes no-inbreeding and the pedigree is deterministic (the individual is offspring of the cross between the two parents). Another example, a population of 10 open-pollinated plants where you harvest the seeds without tracking the mother. 2) When `fixedParent` is TRUE: a mating design where you know one parent and might know the other possible parents. For example, a polycross design where you have seeds harvested from a mother plant and possible pollen donors.

### Usage

```
AmatrixPolyCross(data = NULL, fixedParent = FALSE)
```

### Arguments

<code>data</code>	pedigree data name. Unknown value should be equal 0. See example for construction.
<code>fixedParent</code>	if false, assumes that all the parents are equally possible parents. If true, assumes that the first parental is known and the others are equally possible parents. Default = FALSE.

### Value

Matrix with the relationship between the individuals.

### Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

### Examples

```
#the following pedigree has the id of the individual followed by possible parents
#if 0 is unknown
#the possible parents are filled from left to right
#in the pedigree data frame examples:
#id 1,2,3,4 have unknown parents and are assumed unrelated
#id 5 has three possible parents (1,2,3)
#id 6 has three possible parents (2,3,4)
#id 7 has two parents (deterministic case here, the parents are 3 and 4)
```

```

#id 8 has four possible parents (5,6,7,1)

pedigree = data.frame(id=1:8,
                      parent1 = c(0,0,0,0,1,2,3,5),
                      parent2 = c(0,0,0,0,2,3,4,6),
                      parent3 = c(0,0,0,0,3,4,0,7),
                      parent4 = c(0,0,0,0,0,0,0,1),
                      parent5 = 0)

print(pedigree)

AmatrixPolyCross(pedigree)

#when polyCross is set to be true:
#id 5 is offspring of parent 1 in a deterministic way and two other possible parents (2,3)
#id 6 is offspring of parent 2 in a deterministic way and two other possible parents (3,4)
#id 7 has two parents (deterministic case here, the parents are 3 and 4); as before
#id 8 is offspring of parent 5 in a deterministic way and has three other possible parents (6,7,1)

AmatrixPolyCross(pedigree, fixedParent=TRUE)

```

---

datatreat

*Organizes pedigree data in a chronological way*


---

## Description

This function organizes pedigree data in a chronological way and return 3 lists: i) parental 1 values (numeric); ii) parental 2 values (numeric); iii) real names of the individuals. Also save a .txt file with new pedigree file.

## Usage

```
datatreat(data = NULL, n.max = 50, unk = 0, save = FALSE)
```

## Arguments

data	name of the pedigree data frame. Default=NULL.
n.max	max number of iterations to get the chronological order. Default = 50
unk	the code of the data missing. Default=0.
save	if TRUE, save the genealogy in a .txt file

## Value

list with parental 1, parental 2, and real names of the individuals (key) also saves a txt file with the new chronological pedigree.

**Author(s)**

Rodrigo R Amadeu, <rramadeu@gmail.com>

**Examples**

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

---

filterpedigree	<i>Filter the pedigree to keep only the genealogy of a subset of individuals</i>
----------------	--

---

**Description**

Filter the pedigree to keep only the genealogy of a subset of individuals

**Usage**

```
filterpedigree(inds, data)
```

**Arguments**

inds	vector with strings of individuals to keep their genealogy in the matrix
data	name of the pedigree data frame. Default=NULL.

**Value**

a data frame with pedigree containing the genealogy of the selected individuals

**Author(s)**

Rodrigo R Amadeu, <rramadeu@gmail.com>

**Examples**

```
data(ped.sol)
new.ped.sol = filterpedigree(inds = c("MSW168-2", "W14090-3", "W14090-4"), data=ped.sol)
```

---

formatmatrix	<i>Transform a matrix in 3 columns</i>
--------------	--

---

### Description

Given any square matrix transform it in a 3 columns way (row, column, value) mainly to be used in outsourcing data processing (as ASREML-standalone)

### Usage

```
formatmatrix(  
  data = NULL,  
  save = TRUE,  
  return = FALSE,  
  name = deparse(substitute(data)),  
  round.by = 12,  
  exclude.0 = TRUE  
)
```

### Arguments

data	matrix (nxn).
save	if TRUE save the output in a file. Default=TRUE.
return	if TRUE return the output in a object. Default=FALSE.
name	name of the csv file to be saved. Default=data name.
round.by	select the number of digits after 0 you want in your data. Default = 12
exclude.0	if TRUE, remove all lines equal to zero (ASREML option). Default = TRUE

### Value

a object or a csv file with a table with 3 columns representing the matrix.

### Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

### Examples

```
#Example with random matrix  
data<-matrix(c(1,0.1,0,0.1,1,0,0,0,1.1),3)  
formatmatrix(data=data,save=FALSE,return=TRUE,exclude.0=TRUE)
```

```
#Example with pedigree matrix  
#Reading the example data  
data(ped.mrode)  
#Making Relationship Matrix  
Amrode<-Amatrix(ped.mrode)
```

```
#Inverting the Matrix
Amrode.inv<-solve(Amrode)
#Making the 3 columns format
Amrode.inv.ASREML<-formatmatrix(Amrode,save=FALSE,return=TRUE,exclude.0=TRUE)
#Printing it
Amrode.inv.ASREML
```

---

Gmatrix

---

*Construction of Relationship Matrix G*


---

### Description

Given a matrix (individual x markers), a method, a missing value, and a maf threshold, return a additive or non-additive relationship matrix. For diploids, the methods "Yang" and "VanRaden" for additive relationship matrices, and "Su" and "Vitezica" for non-additive relationship matrices are implemented. For autopolyploids, the method "VanRaden" for additive relationship, method "Slater" for full-autopolyploid model including non-additive effects, and pseudo-diploid parametrization are implemented. Weights are implemented for "VanRaden" method as described in Liu (2020).

### Usage

```
Gmatrix(
  SNPmatrix = NULL,
  method = "VanRaden",
  missingValue = -9,
  maf = 0,
  thresh.missing = 1,
  verify.posdef = FALSE,
  ploidy = 2,
  pseudo.diploid = FALSE,
  integer = TRUE,
  ratio = FALSE,
  impute.method = "mean",
  ratio.check = TRUE,
  weights = NULL,
  ploidy.correction = FALSE
)
```

### Arguments

SNPmatrix	matrix (n x m), where n is individual names and m is marker names (coded inside the matrix as 0, 1, 2, ..., ploidy, and, missingValue).
method	"Yang" or "VanRaden" for marker-based additive relationship matrix. "Su" or "Vitezica" for marker-based dominance relationship matrix. "Slater" for full-autopolyploid model including non-additive effects. "Endelman" for autotetraploid dominant (digentic) relationship matrix. "MarkersMatrix" for a matrix with the amount of shared markers between individuals (3). Default is



	"VanRaden", for autopolyploids will be computed a scaled product (similar to Covarrubias-Pazaran, 2006).
missingValue	missing value in data. Default=-9.
maf	minimum allele frequency accepted to each marker. Default=0.
thresh.missing	threshold on missing data, SNPs below of this frequency value will be maintained, if equal to 1, no threshold and imputation is considered. Default = 1.
verify.posdef	verify if the resulting matrix is positive-definite. Default=FALSE.
ploidy	data ploidy (an even number between 2 and 20). Default=2.
pseudo.diploid	if TRUE, uses pseudodiploid parametrization of Slater (2016).
integer	if FALSE, not check for integer numbers. Default=TRUE.
ratio	if TRUE, molecular data are considered ratios and its computed the scaled product of the matrix (as in "VanRaden" method).
impute.method	"mean" to impute the missing data by the mean per marker, "mode" to impute the missing data by the mode per marker, "global.mean" to impute the missing data by the mean across all markers, "global.mode" to impute the missing data by the mode across all marker. Default = "mean".
ratio.check	if TRUE, run snp.check with ratio data.
weights	vector with weights for each marker. Only works if method="VanRaden". Default is a vector of 1's (equal weight).
ploidy.correction	It sets the denominator (correction) of the crossprod. Used only when ploidy > 2 for "VanRaden" and ratio models. If TRUE, it uses the sum of "Ploidy" times "Frequency" times "(1-Frequency)" of each marker as method 1 in VanRaden 2008 and Endelman (2018). When ratio=TRUE, it uses "1/Ploidy" times "Frequency" times "(1-Frequency)". If FALSE, it uses the sum of the sampling variance of each marker. Default = FALSE.

**Value**

Matrix with the marker-bases relationships between the individuals

**Author(s)**

Rodrigo R Amadeu <rramadeu@gmail.com>, Marcio Resende Jr, Leticia AC Lara, Ivone Oliveira, and Felipe V Ferrao

**References**

- Covarrubias-Pazaran, G. 2016. Genome assisted prediction of quantitative traits using the R package sommer. *PLoS ONE* 11(6):1-15.
- Slater, AT, et al. 2016. Improving genetic gain with genomic selection in autotetraploid potato. *The Plant Genome* 9(3), pp.1-15.
- Su, G, et al. 2012. Estimating additive and non-additive genetic variances and predicting genetic merits using genome-wide dense single nucleotide polymorphism markers. *PloS one*, 7(9), p.e45293.

VanRaden, PM, 2008. Efficient methods to compute genomic predictions. *Journal of dairy science*, 91(11), pp.4414-4423.

Vitezica, ZG, et al. 2013. On the additive and dominant variance and covariance of individuals within the genomic selection scope. *Genetics*, 195(4), pp.1223-1230.

Yang, J, et al. 2010. Common SNPs explain a large proportion of the heritability for human height. *Nature genetics*, 42(7), pp.565-569.

Endelman, JB, et al., 2018. Genetic variance partitioning and genome-wide prediction with allele dosage information in autotetraploid potato. *Genetics*, 209(1) pp. 77-87.

Liu, A, et al. 2020. Weighted single-step genomic best linear unbiased prediction integrating variants selected from sequencing data by association and bioinformatics analyses. *Genet Sel Evol* 52, 48.

## Examples

```
## Not run:
## Diploid Example
data(snp.pine)
#Verifying if data is coded as 0,1,2 and missing value.
str(snp.pine)
#Build G matrices
Gmatrix.Yang <- Gmatrix(snp.pine, method="Yang", missingValue=-9, maf=0.05)
Gmatrix.VanRaden <- Gmatrix(snp.pine, method="VanRaden", missingValue=-9, maf=0.05)
Gmatrix.Su <- Gmatrix(snp.pine, method="Su", missingValue=-9, maf=0.05)
Gmatrix.Vitezica <- Gmatrix(snp.pine, method="Vitezica", missingValue=-9, maf=0.05)

## Autetraploid example
data(snp.sol)
#Build G matrices
Gmatrix.VanRaden <- Gmatrix(snp.sol, method="VanRaden", ploidy=4)
Gmatrix.Endelman <- Gmatrix(snp.sol, method="Endelman", ploidy=4)
Gmatrix.Slater <- Gmatrix(snp.sol, method="Slater", ploidy=4)
Gmatrix.Pseudodiploid <- Gmatrix(snp.sol, method="VanRaden", ploidy=4, pseudo.diploid=TRUE)

#Build G matrix with weights
Gmatrix.weighted <- Gmatrix(snp.sol, method="VanRaden", weights = runif(3895,0.001,0.1), ploidy=4)

## End(Not run)
```

## Description

Given a matrix A and a matrix G returns a H matrix. H matrix is the relationship matrix using combined information from the pedigree and genomic relationship matrices. First, you need to compute the matrices separated and then use them as input to build the combined H matrix. Two methods are

implemented: 'Munoz' shrinks the G matrix towards the A matrix scaling the molecular relationship by each relationship classes; 'Martini' is a modified version from Legarra et al. (2009) where combines A and G matrix using scaling factors. When method is equal 'Martini' and 'tau=1' and 'omega=1' you have the same H matrix as in Legarra et al. (2009).

### Usage

```
Hmatrix(
  A = NULL,
  G = NULL,
  markers = NULL,
  c = 0,
  method = "Martini",
  tau = 1,
  omega = 1,
  missingValue = -9,
  maf = 0,
  ploidy = 2,
  roundVar = 3
)
```

### Arguments

A	A matrix from function Amatrix
G	G matrix from function Gmatrix
markers	matrix marker which generated the Gmatrix
c	constant value of H computation, default: c=0
method	"Martini" or "Munoz", default="Martini"
tau	to be used for Martini's method, default=1.
omega	to be used of Martini's method, default=1.
missingValue	missing value in data, default=-9.
maf	max of missing data accepted to each marker default=0.05.
ploidy	data ploidy (an even number between 2 and 20), default=2.
roundVar	only used for Munoz's method, how many digits to consider the relationship between of same class, default=2.

### Value

H Matrix with the relationship between the individuals based on pedigree and corrected by molecular information

### Author(s)

Rodrigo R Amadeu, <rramadeu@gmail.com>

## References

- Munoz, PR. 2014 Unraveling additive from nonadditive effects using genomic relationship matrices. *Genetics* 198, 1759-1768
- Martini, JW, et al. 2018 The effect of the H-1 scaling factors tau and omega on the structure of H in the single-step procedure. *Genetics Selection Evolution* 50(1), 16
- Legarra, A, et al. 2009 A relationship matrix including full pedigree and genomic information. *Journal of Dairy Science* 92, 4656–4663

## Examples

```
## Not run:
data(ped.sol)
data(snp.sol)
#Computing the numerator relationship matrix 10% of double-reduction
Amat <- Amatrix(ped.sol, ploidy=4, w = 0.1)
#Computing the additive relationship matrix based on VanRaden (modified)
Gmat <- Gmatrix(snp.sol, ploidy=4,
               maf=0.05, method="VanRaden")
Gmat <- round(Gmat,3) #to be easy to invert

#Computing H matrix (Martini)
Hmat_Martini <- Hmatrix(A=Amat, G=Gmat, method="Martini",
                       ploidy=4,
                       maf=0.05)

#Computing H matrix (Munoz)
Hmat_Munoz <- Hmatrix(A=Amat, G=Gmat, markers = snp.sol,
                      ploidy=4, method="Munoz",
                      roundVar=2,
                      maf=0.05)

## End(Not run)
```

---

missingdata

*Survying on missing data*

---

## Description

This function verify which rows in a pedigree data has missing parental or conflictuos data

## Usage

```
missingdata(data, unk = 0)
```

## Arguments

data	data name from a pedigree list
unk	unknown value of your data

**Value**

list with \$conflict: rows of the data which are at least one parental name equal to the individual. \$missing.sire: rows of the data which are missing data sire (Parental 1) information. \$missing.dire: same as above for dire (Parental 2). \$summary.missing: summary of the missing data. 2 columns, 1st for the name of the parental listed, 2nd for the how many times appeared in the data.

**Author(s)**

Rodrigo R Amadeu, <rramadeu@gmail.com>

**Examples**

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

---

ped.mrode

*Pedigree Data*

---

**Description**

Data from pedigree example proposed by Mrode 2005

**Usage**

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

**Format**

table

**References**

R. A. Mrode, R. Thompson. Linear Models for the Prediction of Animal Breeding Values. CABI, 2005.

**Examples**

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

---

`ped.sol`*Pedigree data for autopolyploid examples*

---

**Description**

Dataset extract from supplementary material from Endelman et al. (2018). Pedigree data frame of Potato population, missing data as 0.

**Usage**

```
data(ped.sol)
```

**Format**

```
data.frame
```

**References**

Endelman, JB, et al., 2018 Genetic variance partitioning and genome-wide prediction with allele dosage information in autotetraploid potato. *Genetics*, 209(1) pp. 77-87.

**Examples**

```
data(ped.sol)
```

---

`snp.pine`*Molecular data for diploid examples*

---

**Description**

Dataset extract from supplementary material from Resende et al. (2012). SNP marker matrix from Pine tree coded as 0,1, and 2, and missing value as -9.

**Usage**

```
data(snp.pine)
```

**Format**

```
matrix
```

**References**

Resende, MF, et al., 2012 Accuracy of genomic selection methods in a standard data set of loblolly pine (*Pinus taeda* L.). *Genetics* 190: 1503–1510.

**Examples**

```
data(snp.pine)
```

---

```
snp.sol
```

*Molecular data for autopolyploid examples*

---

**Description**

Dataset extract from supplementary material from Endelman et al. (2018). SNP marker matrix from Pine tree coded as 0,1,2,3,4 and missing value as -9.

**Usage**

```
data(snp.sol)
```

**Format**

```
data.frame
```

**References**

Endelman, JB, et al., 2018 Genetic variance partitioning and genome-wide prediction with allele dosage information in autotetraploid potato. *Genetics*, 209(1) pp. 77-87.

**Examples**

```
data(snp.sol)
```

# Index

## \* datasets

ped.mrode, 13

ped.sol, 14

snp.pine, 14

snp.sol, 15

Amatrix, 2

AmatrixPolyCross, 4

datatreat, 5

filterpedigree, 6

formatmatrix, 7

Gmatrix, 8

Hmatrix, 10

missingdata, 12

ped.mrode, 13

ped.sol, 14

snp.pine, 14

snp.sol, 15