

# Package ‘pedigreeTools’

December 9, 2018

**Version** 0.1

**Date** 2018-11-14

**Title** Versatile Functions for Working with Pedigrees

**Author** Ana Ines Vazquez, Douglas Bates, Siddharth Avadhanam, Paulino Perez

**Maintainer** Ana Ines Vazquez <avazquez@msu.edu>

**Description** Tools to sort, edit and prune pedigrees and to extract the inbreeding coefficients and the relationship matrix (includes code for pedigrees from self-pollinated species). The use of pedigree data is central to genetics research within the animal and plant breeding communities to predict breeding values. The relationship matrix between the individuals can be derived from pedigree structure following the algorithms described for example in Vazquez et al., 2010 <doi:10.2527/jas.2009-1952>.

**Depends** R(>= 3.0.0), methods

**Imports** Matrix (>= 1.0)

**LazyLoad** yes

**LazyData** yes

**License** GPL-3

**URL** <https://github.com/Rpedigree/pedigreeTools/>

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-12-09 13:50:06 UTC

## R topics documented:

Dmat	2
editPed	2
getA	3
getAInv	4
getASelfing	4

getGenAncestors . . . . .	5
inbreeding . . . . .	6
ped2DF . . . . .	6
pedigree . . . . .	7
pedigree-class . . . . .	7
prunePed . . . . .	8
refactor . . . . .	8

## Index 10

---

Dmat	<i>Diagonal of D in the <math>A = TDT'</math> factorization.</i>
------	--

---

### Description

Determine the diagonal factor in the decomposition of the relationship matrix  $A$  as  $TDT'$  where  $T$  is unit lower triangular.

### Usage

```
Dmat(ped)
```

### Arguments

ped                    an object that inherits from class [pedigree](#)

### Value

a numeric vector

### Examples

```
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
Dmat(ped)
```

---

editPed	<i>Edits a disordered or incomplete pedigree.</i>
---------	---

---

### Description

1\_ add labels for the sires and dams not listed as labels before. 2\_ order pedigree based on recursive calls to getGenAncestors.

### Usage

```
editPed(sire, dam, label, verbose = FALSE)
```

**Arguments**

sire	integer vector or factor representation of the sires
dam	integer vector or factor representation of the dams
label	character vector of labels
verbose	logical to print the row of the pedigree that the function is ordering. Default is FALSE.

**Value**

a data frame with the pedigree ordered.

**Examples**

```
pede <- data.frame(sire=as.character(c(NA,NA,NA,NA,NA,1,3,5,6,4,8,1,10,8)),
                  dam=as.character(c(NA,NA,NA,NA,NA,2,2,NA,7,7,NA,9,9,13)),
                  label=as.character(1:14))
pede <- pede[sample(replace=FALSE, 1:14),]
pede <- editPed(sire=pede$sire, dam=pede$dam, label=pede$label)
ped <- with(pede, pedigree(label=label, sire=sire, dam=dam))
```

---

getA

*Additive Relationship Matrix*


---

**Description**

Returns the additive relationship matrix for the pedigree ped.

**Usage**

```
getA(ped)
```

**Arguments**

ped	a pedigree that includes the individuals who occur in svec which to restrict the relationship matrix. If labs is a factor then the levels of the factor are used as the labels. Default is the complete set of labels in the pedigree.
-----	--

**Value**

an object that inherits from [CHMfactor](#)

**Examples**

```
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
getA(ped)
```

---

getAInv	<i>Inverse of the Relationship Matrix</i>
---------	---

---

**Description**

Inverse of the Relationship Matrix

**Usage**

```
getAInv(ped)
```

**Arguments**

ped	a pedigree that includes the individuals who occur in svec which to restrict the relationship matrix. If labs is a factor then the levels of the factor are used as the labels. Default is the complete set of labels in the pedigree.
-----	--

**Value**

an object that inherits from [CHMfactor](#)

**Examples**

```
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
getAInv(ped)
```

---

getASelfing	<i>Extends the pedigree according to number of selfing cycles and also optionally computes the Additive Relationship Matrix for that pedigree.</i>
-------------	--

---

**Description**

Extends the pedigree according to number of selfing cycles and also optionally computes the Additive Relationship Matrix for that pedigree.

**Usage**

```
getASelfing(ID, Par1, Par2, nCycles, nCyclesDefault, sepChar = "-F",
  verbose = FALSE, fileNewPed = NULL, computeA = TRUE)
```

**Arguments**

ID	is a vector of individual IDs
Par1	vector of IDs of one of the parents
Par2	vector of IDs of the other parent
nCycles	vector that indicates number of selfing cycles for each individual.
nCyclesDefault	default value of nCycles
sepChar	sepChar
verbose	print progress
fileNewPed	Output csv file (comma separated value) with columns 'label', 'sire', 'dam', with the full pull pedigree expanded taking into account the selfing cycles
computeA	Indicates if the A matrix is to be computed

**Value**

Returns A matrix computed for the extended pedigree if computeA=TRUE

---

getGenAncestors	<i>Counts number of generations of ancestors for one subject. Use recursion.</i>
-----------------	--

---

**Description**

Counts number of generations of ancestors for one subject. Use recursion.

**Usage**

```
getGenAncestors(pede, id, ngen = NULL)
```

**Arguments**

pede	data frame with a pedigree and a column for the number of generations of each subject.
id	subject for which we want the number of generations.
ngen	number of generation

**Value**

a data frame object with the pedigree and generation of ancestors for subject id.

---

inbreeding	<i>Inbreeding coefficients from a pedigree</i>
------------	--

---

**Description**

Create the inbreeding coefficients according to the algorithm given in "Comparison of four direct algorithms for computing inbreeding coefficients" by Mehdi Sargolzaei and Hiroaki Iwaisaki, Animal Science Journal (2005) 76, 401–406.

**Usage**

```
inbreeding(ped)
```

**Arguments**

ped                    an object that inherits from class [pedigree](#)

**Value**

the inbreeding coefficients as a numeric vector

**Examples**

```
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
inbreeding(ped)
```

---

ped2DF	<i>Convert a pedigree to a data frame</i>
--------	---

---

**Description**

Express a pedigree as a data frame with sire and dam stored as factors. If the pedigree is an object of class `pedinbred` then the inbreeding coefficients are appended as the variable `F`

**Usage**

```
ped2DF(x)
```

**Arguments**

x                    a pedigree object of class [pedigree](#)

**Value**

a data frame

---

pedigree	<i>Constructor for pedigree objects</i>
----------	---

---

**Description**

A simple constructor for a pedigree object. The main point for the constructor is to use coercions to make the calls easier.

**Usage**

```
pedigree(sire, dam, label)
```

**Arguments**

sire	integer vector or factor representation of the sires
dam	integer vector or factor representation of the dams
label	character vector of labels

**Value**

an pedigree object of class [pedigree](#)

**Note**

sire, dam and label must all have the same length and all labels in sire and dam must occur in label

**Examples**

```
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
```

---

pedigree-class	<i>Pedigree class</i>
----------------	-----------------------

---

**Description**

Pedigree class

---

prunePed	<i>Subsets a pedigree for a specified vector of individuals upto a specified number of previous generations using Recursion.</i>
----------	--

---

**Description**

Subsets a pedigree for a specified vector of individuals upto a specified number of previous generations using Recursion.

**Usage**

```
prunePed(ped, selectVector, ngen = 2)
```

**Arguments**

ped	Data Frame pedigree to be subset
selectVector	Vector of individuals to select from pedigree
ngen	Number of previous generations of parents to select starting from selectVector.

**Value**

Returns Subsetted pedigree as a DataFrame.

---

relfactor	<i>Relationship factor from a pedigree</i>
-----------	--

---

**Description**

Determine the right Cholesky factor of the relationship matrix for the pedigree ped, possibly restricted to the specific labels that occur in labs.

**Usage**

```
relfactor(ped, labs)
```

**Arguments**

ped	a pedigree that includes the individuals who occur in svec
labs	a character vector or a factor giving the labels to which to restrict the relationship matrix. If labs is a factor then the levels of the factor are used as the labels. Default is the complete set of labels in the pedigree.

**Value**

an object that inherits from [CHMfactor](#)

**Examples**

```
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
refactor(ped)
```

# Index

\*Topic **array**

getA, [3](#)

CHMfactor, [3](#), [4](#), [8](#)

Dmat, [2](#)

editPed, [2](#)

getA, [3](#)

getAInv, [4](#)

getASelfing, [4](#)

getGenAncestors, [5](#)

inbreeding, [6](#)

ped2DF, [6](#)

pedigree, [2](#), [6](#), [7](#), [7](#)

pedigree-class, [7](#)

prunePed, [8](#)

relfactor, [8](#)