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
\textbf{dmat} \hspace{1cm} \textit{Diagonal of D in the A = TDT' factorization.}

**Description**

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

**Usage**

\begin{verbatim}
dmatHpedI
\end{verbatim}

**Arguments**

\begin{verbatim}
ped \hspace{1cm} \text{an object that inherits from class \texttt{pedigree}}
\end{verbatim}

**Value**

a numeric vector

**Examples**

\begin{verbatim}
ped <- pedigree(sire=c(NA,NA,1,1,4,5), dam=c(NA,NA,2,NA,3,2), label=1:6)
Dmat(ped)
\end{verbatim}

\begin{verbatim}
editPedI \hspace{1cm} \textit{Edits a disordered or incomplete pedigree.}
\end{verbatim}

**Description**

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

**Usage**

\begin{verbatim}
editPed(sire, dam, label, verbose = FALSE)
\end{verbatim}
getA

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

```r
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,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)
pede <- 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

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

Inverse of the Relationship Matrix

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)

gerAASelfing

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

Description

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

Usage

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

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

**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

Inbreeding coefficients from a pedigree

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

Convert a pedigree to a data frame

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
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)
prunePed: Subsets a pedigree for a specified vector of individuals up to a specified number of previous generations using Recursion.

**Description**
Subsets a pedigree for a specified vector of individuals up to a specified number of previous generations using Recursion.

**Usage**
```r
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.

---

def relfactor(ped, labs)

**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**
```r
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

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

relnfactor, 8