

Package ‘pedigreemm’

February 15, 2013

Version 0.2-4

Date 2009-05-29

Title Pedigree-based mixed-effects models

Author Douglas Bates and Ana Ines Vazquez

Maintainer Ana Ines Vazquez <anainesvs@gmail.com>

Description Fit pedigree-based mixed-effects models.

Depends methods, R(>= 2.8.1), lme4 (>= 0.999375-28), Matrix (>= 0.999375-22)

Imports graphics, stats

LazyLoad yes

LazyData yes

License GPL (>= 2)

URL <http://pedigreemm.r-forge.r-project.org/>

Repository CRAN

Repository/R-Forge/Project pedigreemm

Repository/R-Forge/Revision 107

Date/Publication 2009-08-13 08:59:08

NeedsCompilation yes

R topics documented:

Dmat	2
inbreeding	3
mastitis	3
milk	4
pedCows	5
pedCowsR	6
pedigree	6
pedigree-class	7
pedigreemm	8
pedigreemm-class	9
pedSires	10
refactor	11
Index	12

Dmat	<i>vector of the diagonal for the D matrix from the decomposition $A = TDT'$</i>
------	---

Description

numeric vector that should be the diagonal elements of the diagonal matrix D

Usage

```
Dmat(ped)
```

Arguments

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

Details

Determine the diagonal factor in the decomposition of the relationship matrix from a pedigree equal to TDT' . Where T is unit lower triangular and D is a diagonal matrix. This function returns a numeric vector with the entries of D

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)
```

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

Description

Inbreeding coefficients from a pedigree

Usage

```
inbreeding(ped)
```

Arguments

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

Details

Determine the inbreeding coefficients for all the individuals of a pedigree. This function a numeric vector.

Value

a numeric vector

Source

Sargolzaei, M. and H. Iwaisaki, 2005. Comparison of four direct algorithms for computing the inbreeding coefficients. *J. Anim. Sci*, 76: 401-406.

Examples

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

mastitis	<i>Mastitis cases in dairy cattle</i>
----------	---------------------------------------

Description

Records of the number of cases of clinical mastitis during the first lactation of 1,675 cows, primarily Holsteins. Cows belonged to 41 herds and were daughters of 38 sires. There were 1,491 healthy cows, 134 had only one case of mastitis, 36 had 2 cases, and 14 had between 4 and cases. Overall, mastitis incidence was 0.11. Calving years for these records were from 2000 through 2005. The sire, herd and days in milk are also recorded for each cow.

Format

A data frame with 1675 observations on the following 8 variables.

`id` Identifier of the animal.

`sire` Identifier of the animal's sire.

`birth` year of birth of the animal (as a factor).

`herd` herd id number (as a factor).

`calvingYear` year of calving for this lactation.

`DIM` total number of days in milk for the lactation.

`mastitis` a factor indicating if the cow had any incidents of clinical mastitis during the lactation.

`NCM` An ordered factor giving the number of clinical mastitis cases for the cow during this lactation.

Details

The pedigree of the sires is given in the companion [pedSires](#) data set.

Source

Vazquez, A.I. 2007. Analysis of number of episodes of clinical mastitis in Norwegian Red and Holstein cows with Poisson and categorical data mixed models. Master of Science Thesis. University of Wisconsin - Madison. 162 pp.

See Also

[pedSires](#), [pedigree](#)

Examples

```
str(mastitis)
summary(mastitis, maxsum = 10)
```

milk

Milk production

Description

Records of the milk production of 3397 lactations from first through fifty parity Holsteins. These were 1,359 cows, daughters of 38 sires in 57 herds. The data was downloaded from the USDA internet site. All lactation records represent cows with at least 100 days in milk, with an average of 347 days. Milk yield ranged from 4,065 to 19,345 kg estimated for 305 days, averaging 11,636 kg. There were 1,314, 1,006, 640, 334 and 103 records were from first thorough fifth lactation animals.

Usage

```
data(milk)
```

Format

A data frame with 3397 observations on the following 9 variables.

id numeric identifier of cow
 lact number of lactation for which production is measured
 herd a factor indicating the herd
 sire a factor indicating the sire
 dim number of days in milk for that lactation
 milk milk production estimated at 305 days
 fat fat production estimated at 305 days
 prot protein production estimated at 305 days
 scs the somatic cell score

Source

USDA web site. <http://www.aipl.arsusda.gov/>

Examples

```
str(milk)
```

pedCows

Pedigree of the cows in milk

Description

A [pedigree](#) object giving (part of) the pedigree of the cows in the [milk](#) data frame.

Format

The format is: Formal class 'pedigree' [package "pedigreemm"] with 3 slots ..@ sire : int [1:6547]
 NA NA NA NA NA NA NA NA NA NA NA@ dam : int [1:6547] NA NA NA NA NA NA NA NA
 NA NA@ label: chr [1:6547] "1" "2" "3" "4" ...

See Also

[milk](#)

Examples

```
str(pedCows)
```

pedCowsR

Pedigree of the cows in milk with 0.70 of the information in pedCows

Description

A [pedigree](#) object giving (part of) the pedigree of the cows in the [milk](#) data frame. This pedigree allows the example with 'milk' to run faster.

Format

The format is: Formal class 'pedigree' [package "pedigreemm"] with 3 slots ..@ sire : int [1:6547] NA NA NA NA NA NA NA NA NA@ dam : int [1:6547] NA NA NA NA NA NA NA NA NA@ label: chr [1:6547] "1" "2" "3" "4" ...

See Also

[milk](#)

Examples

```
str(pedCowsR)
```

pedigree

Pedigree Constructor

Description

Construct an object of class "pedigree", more conveniently than by `new("pedigree", ...)`.

Usage

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

Arguments

sire	numeric vector (with some NA entries) of integer IDs, denoting a <i>previous</i> entry in the pedigree corresponding to the current entry's "father".
dam	similarly as sire for the "mother" of each entry.
label	a vector coercable to "character" of the same length as sire and dam giving a unique ID for the corresponding entry.

Value

an object of formal class "[pedigree](#)".

See Also

the [pedigree](#) class.

Examples

```
example("pedigree-class") ## 'p1' pedigree object 'the hard way'

ped <- pedigree(sire = c(NA,NA,1, 1,4,5),
               dam  = c(NA,NA,2,NA,3,2), label= 1:6)
## note that 'label' is coerced to character automatically
ped
stopifnot(identical(ped, p1))
```

pedigree-class	<i>Class "pedigree"</i>
----------------	-------------------------

Description

Objects of class "pedigree" represent a set of individuals that can have two parents including their parent-child relations. The terminology has been taken from cattle breeding. The "pedinbred" class is an extension of the pedigree class with an additional slot of the inbreeding coefficients.

Objects from the Class

Objects in the "pedigree" class can be created by calls of the form `new("pedigree", ...)`, or more conveniently, `pedigree(sire= ., dam = ., label =.)`.

Objects of the "pedinbred" class are created by coercing a pedigree to class "pedinbred".

Slots

sire: integer vector (with some NA entries), denoting a *previous* entry in the pedigree corresponding to the current entry's "father".

dam: similarly as sire for the "mother" of each entry.

label: a "character" vector of the same length as sire and dam giving a unique ID for the corresponding entry.

F: (class "pedinbred" only) a numeric vector of inbreeding coefficients.

Methods

coerce signature(from = "pedigree", to = "sparseMatrix"): returns a sparse, unit lower-triangular matrix which is the inverse of the "L" part of the "LDL" form of the Cholesky factorization of the relationship matrix. All non-zero elements below the diagonal are -0.5.

coerce signature(from = "pedigree", to = "data.frame"): ...

head signature(x = "pedigree"): ...

show signature(object = "pedigree"): ...

tail signature(x = "pedigree"): ...

References

R. A. Mrode, *Linear Models for the Prediction of Animal Breeding Values*, 2nd ed, CABI Publishing, 2005.

See Also

[pedigree](#), [inbreeding](#)

Examples

```
## Rather use, pedigree()! The following is "raw code":
## Example from chapter 2 of Mrode (2005)
p1 <- new("pedigree",
         sire = as.integer(c(NA,NA,1, 1,4,5)),
         dam  = as.integer(c(NA,NA,2,NA,3,2)),
         label = as.character(1:6))

p1
(dtc <- as(p1, "sparseMatrix")) # T-inverse in Mrode's notation
solve(dtc)
inbreeding(p1)
```

pedigreemm

Fit mixed-effects models incorporating pedigrees

Description

Fit linear or generalized linear mixed models incorporating the effects of a pedigree.

Usage

```
pedigreemm(formula, data, family = NULL, REML = TRUE,
           pedigree = list(), control = list(),
           start = NULL, verbose = FALSE, subset, weights,
           na.action, offset, contrasts = NULL, model = TRUE,
           x = TRUE, ...)
```

Arguments

pedigree	a named list of pedigree objects. The names must correspond to the names of grouping factors for random-effects terms in the formula argument.
formula	as in lmer
data	as in lmer
family	as in lmer
REML	as in lmer
control	as in lmer
start	as in lmer

verbose	as in lmer
subset	as in lmer
weights	as in lmer
na.action	as in lmer
offset	as in lmer
contrasts	as in lmer
model	as in lmer
x	as in lmer
...	as in lmer

Details

All arguments to this function are the same as those to the function [lmer](#) except pedigree which must be a named list of [pedigree](#) objects. Each name (frequently there is only one) must correspond to the name of a grouping factor in a random-effects term in the formula. The observed levels of that factor must be contained in the pedigree. For each pedigree the (left) Cholesky factor of the relationship matrix restricted to the observed levels is calculated using [relfactor](#) and applied to the model matrix for that term.

Value

a [pedigreemm](#) object.

See Also

[pedigreemm](#), [pedigree](#), [relfactor](#).

pedigreemm-class	<i>Pedigree-based mixed-effects model fits</i>
------------------	--

Description

A mixed-effects model fit by [pedigreemm](#). This class extends class "[mer](#)" class and includes one additional slot, `relfac`, which is a list of (left) Cholesky factors of the relationship matrices derived from "[pedigree](#)" objects.

Objects from the Class

Objects are created by calls to the [pedigreemm](#) function.

Slots

`relfac`: A list of relationship matrix factors. All other slots are inherited from class "[mer](#)".

`env`, `nlmodel`, `frame`, `call`, `flist`, `X`, `Zt`, `pWt`, `offset`, `y`, `Gp`, `dims`, `ST`, `V`, `A`, `Cm`, `Cx`, `L`, `deviance`, `fixef`, `ranef`, `u`, `eta`, `mu`,
all other slots are inherited from the superclass "[mer](#)" in the `lme4` package.

Extends

Class "[mer](#)", directly.

Methods

No methods defined with class "pedigreemm" in the signature.

See Also

[pedigreemm](#)

Examples

```
showClass("pedigreemm")
```

pedSires

Pedigree of the sires from mastitis

Description

A [pedigree](#) object giving (part of) the pedigree of the sires from the [mastitis](#) data frame. The pedigree is traced back on sires only.

Format

The format is: Formal class 'pedigree' [package "pedigreemm"] with 3 slots ..@ sire : int [1:352]
NA NA NA NA NA NA NA NA NA NA NA NA@ dam : int [1:352] NA NA NA NA NA NA NA NA NA
NA NA@ label: chr [1:352] "1" "2" "3" "4" ...

See Also

[mastitis](#)

Examples

```
str(pedSires)
```

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

Description

Relationship factor from a pedigree

Usage

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

Arguments

ped	a pedigree that includes the individuals who occur in labs
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.

Details

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

Value

an upper triangular, sparse (right) Cholesky factor of the relationship matrix

Examples

```
## Example from chapter 2 of Mrode (2005)

ped <- pedigree(sire = c(NA,NA,1, 1,4,5),
               dam  = c(NA,NA,2,NA,3,2), label= 1:6)
(fac <- relfactor(ped))
crossprod(fac) # the relationship matrix
```

Index

- *Topic **algebra**
 - relfactor, 11
 - *Topic **array**
 - relfactor, 11
 - *Topic **classes**
 - pedigree-class, 7
 - pedigreemm-class, 9
 - *Topic **datasets**
 - mastitis, 3
 - milk, 4
 - pedCows, 5
 - pedCowsR, 6
 - pedSires, 10
 - *Topic **misc**
 - Dmat, 2
 - inbreeding, 3
 - pedigree, 6
 - *Topic **models**
 - pedigreemm, 8
- coerce, pedigree, data.frame-method
(pedigree-class), 7
- coerce, pedigree, sparseMatrix-method
(pedigree-class), 7
- Dmat, 2
- head, pedigree-method (pedigree-class), 7
- inbreeding, 3, 8
- lmer, 8, 9
- mastitis, 3, 10
- mer, 9, 10
- milk, 4, 5, 6
- pedCows, 5
- pedCowsR, 6
- pedigree, 2–6, 6, 7–10
- pedigree-class, 7
- pedigreemm, 8, 9, 10
- pedigreemm-class, 9
- pedinbred-class (pedigree-class), 7
- pedSires, 4, 10
- relfactor, 9, 11
- show, pedigree-method (pedigree-class), 7
- tail, pedigree-method (pedigree-class), 7