Package 'MasterBayes'

October 12, 2022

Title ML and MCMC Methods for Pedigree Reconstruction and Analysis

Version 2.58

Depends coda, genetics, gtools, kinship2

Imports methods, stats

Date 2022-06-22

Author Jarrod Hadfield

Maintainer Jarrod Hadfield <j.hadfield@ed.ac.uk>

Description The primary aim of 'MasterBayes' is to use MCMC techniques to integrate over uncertainty in pedigree configurations estimated from molecular markers and phenotypic data. Emphasis is put on the marginal distribution of parameters that relate the phenotypic data to the pedigree. All simulation is done in compiled 'C++' for efficiency.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-06-22 13:00:10 UTC

R topics documented:

`autocorrP` *Autocorrelation Function for Parenatge Assignment*

Description

Function for assessing mixing of the Markov chain with respect to parentage assignment.

Usage

```
autocorrP(postP)
```

Arguments

`postP`	JOINT posterior distribution of parentage

Details

For each offspring the proportion of transitions is calculated at lags 1, 2, 5, 10, 50 and 100 (i.e. the proportion of times that the parentage assignment at time t is different from the parentage assignment at time t+lag). The difference between these proportions and the proportion at lag 1 is then calculated, and the mean over offspring given. When the parentage assignments in successive MCMC iterations are independent these autocorrelation metrics should be randomly distributed about zero and should not decrease with increasing lag.

Value

matrix

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)
GdP<-GdataPed(WarblerG)

var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))

# paternity is to be modelled as a function of distance
# between offspring and male territories

res1<-expression(varPed("offspring", restrict=0))

# individuals from the offspring generation are excluded as parents

res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

# mothers not from the offspring territory are excluded

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
tP<-tunePed(beta=30)

model1<-MCMCped(PdP=PdP, GdP=GdP, tP=tP, nitt=3000, thin=1, burnin=0, write_postP="JOINT")
autocorrP(model1$P)

## End(Not run)
```

---

beta.loglik *Log-Likelihood of Beta*

Description

Log-likelihood of beta given a pedigree and phenotypic data. Beta is the parameter vector for the multinomial log-linear model. Intended to be used within the function MLE.beta

Usage

```
beta.loglik(X, dam_pos=NULL, sire_pos=NULL, par_pos=NULL, beta=NULL,
   beta_map=NULL, merge=NULL, mergeN=NULL, nUS=c(0,0), shrink=NULL)
```

Arguments

X	list of design matrices for each offspring. Each element should either have dam (D) and/or sire (S) matrices, or a composite Dam/Sire (DS) matrix. See varPed for model types
dam_pos	position of each offspring's mother in the dam design matrix
sire_pos	position of each offspring's mother in the sire design matrix
par_pos	position of each offspring's parents in the composite dam/sire matrix
beta	parameter vector
beta_map	vector that maps beta onto the design matrices (see getXlist)
merge	optional vector that indicates columns of for which the parameter is transformed using the argument merge in varPed
mergeN	optional list of matrices for each offspring the columns of which refer to merged variables and the rows to the number of individuals that fall into each category defined by merge)
nUS	vector of the number of unsampled females and males, respectively. Only required if unsampled individuals have known phenotype.
shrink	optional scalar for the variance defining the ridge-regression likelihood penalisation.

Value

log-likelihood of beta given the pedigree and X.

Note

Intended to be used within MLE.beta

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31 Smouse P.E. *et al* (1999) Journal of Evolutionary Biology 12 1069-1077

See Also

MLE.beta, MCMCped, varPed, getXlist

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)

GdP<-GdataPed(WarblerG)
```

```
res1<-expression(varPed("offspring", relational=FALSE, restrict=0))
var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))
res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP)

# probability of paternity is modelled as a function of  distance

X.list<-getXlist(PdP=PdP, GdP=GdP)

ped<-MLE.ped(X.list)$P

# get ML pedigree from genetic data alone

X<-lapply(X.list$X, function(x){list(S=x$XSs)})

# Extract Design matrices for Sires

sire_pos<-match(ped[,3][as.numeric(names(X))], X.list$id)
sire_pos<-mapply(function(x,y){match(x, y$sire.id)}, sire_pos, X.list$X)

# row number of each design matrix corresponding to the ML sire.

beta<-seq(-0.065,-0.0325, length=100)
beta_Loglik<-1:100
  for(i in 1:100){
     beta_Loglik[i]<-beta.loglik(X, sire_pos=sire_pos, beta=beta[i],
     beta_map=X.list$beta_map)
  }

plot(beta_Loglik~beta, type="l", main="Profile Log-likelihood for beta")

## End(Not run)
```

---

consensusG *Obtains a consensus genotype from duplicate samples*

Description

A function for obtaining a consensus genotype from duplicate samples. The amount of missing data is minimised, and preference is given to samples with lower genotyping error

Usage

```
consensusG(GdP, cat.levels=NULL, gmax=FALSE, het=FALSE)
```

Arguments

GdP	a GdataPed object
cat.levels	order of genotyping error rate categories, with most reliable category first
gmax	logical; if a most represented genotype exists should it be saved
het	logical; should heterozygotes be saved over homozygotes - overrides cat.levels

Value

GdP	a GdataPed object

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

GdataPed

---

extractA *Allele Frequencies*

Description

extracts allele frequencies from genotype data

Usage

```
extractA(G, marker.type="MSW")
```

Arguments

G	data frame or list of genotype objects
marker.type	"MSW" or "MSC" for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Marshall, 1998) or "AFLP" for dominant markers.

Value

list of allele frequnecies at each loci

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

genotype.list, genotype

Examples

```
## Not run:
  data(WarblerG)

  A<-extractA(WarblerG)
  A[[1]]

## End(Not run)
```

`fillX.G` *Mendelian Transition Probabilities*

Description

This function is primarily intended for use within `getXlist`, and fills in the design matrices of the model with the genetic likelihoods.

Usage

```
fillX.G(X.list, A, G, E1=0.005, E2=0.005, marker.type="MSW")
```

Arguments

`X.list`	list of design matrices for each offspring derived using `getXlist`
`A`	list of allele frequencies
`G`	list of genotype objects; rows must correspond to individuals in the vector `X.list$id`
`E1`	if Wang's (2004) model of genotyping error for co-dominant markers is used this is the probability of an allele dropping out. If CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers is used this parameter is not used. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a dominant allele being scored as a recessive allele.
`E2`	if Wang's (2004) or CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers are used this is the probability of an allele being miss-scored. In the CERVUS model errors are not independent for the two alleles within a genotype and so if a genotyping error has occurred at one allele then a genotyping error occurs at the other allele with probability one. Accordingly, `E2`(2-`E2`) is the per-genotype rate defined in CERVUS. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a recessive allele being scored as a dominant allele.
`marker.type`	`"MSW"` or `"MSC"` for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Kalinowski, 2006; Marshall, 1998) or `"AFLP"` for dominant markers (Hadfield, 2009).

Value

list of design matrices of the form `X.list` containing genetic likelihoods for each offspring.

Note

If a GdataPed object is passed to getXlist then the genetic likelihoods will be calculated by default.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Marshall, T. C. *et al* (1998) Molecular Ecology 7 5 639-655 Kalinowski S.T. *et al* (2007) Molecular Ecology 16 5 1099-1106 Hadfield J. D. *et al* (2009) *in prep*

See Also

getXlist

Examples

```
## Not run:
data(WarblerG)
A<-extractA(WarblerG)

ped<-matrix(NA, 5,3)
ped[,1]<-1:5
ped[,2]<-c(rep(NA, 4), 1)
ped[,3]<-c(rep(NA, 4), 2)

genotypes<-simgenotypes(A, ped=ped)

sex<-c("Female", "Male", "Female", "Male","Female")
offspring<-c(0,0,0,0,1)

data<-data.frame(id=ped[,1], sex, offspring)

res1<-expression(varPed(x="offspring", restrict=0))

PdP<-PdataPed(formula=list(res1), data=data)
GdP<-GdataPed(G=genotypes$Gobs, id=genotypes$id)

X.list<-getXlist(PdP)
# creates design matrices for offspring (in this case indivdiual "5")

X.list.G<-fillX.G(X.list, A=A, G=genotypes$Gobs, E2=0.005)
# genetic likelihoods are arranged sires within dams

X.list.G$X$"5"$dam.id
X.list.G$X$"5"$sire.id

# so for this example we have parental combinations
# ("1","2"), ("1","4"), ("3","2"), ("2","4"):
```

```
X.list.G$X$"5"$G

# The true parents have the highest likelihood in this case

## End(Not run)
```

GdataPed *GdataPed Object*

Description

An object containing genotype data and the categories over which error rates may vary.

Usage

```
GdataPed(G, id = NULL, categories = NULL, perlocus=FALSE, marker.type="MSW")
```

Arguments

G	a list of genotype objects for each locus, or a data.frame to be coerced using genotype.list
id	a vector of individual identifiers associated with each genotype, individuals can have more than one observed genotype. If G is a data.frame to be coerced and has a column name id, this will be used.
categories	an optional vector indicating subsets of genotypes that have different error rates. If G is a data.frame to be coerced and has a column name categories, this will be used.
perlocus	if TRUE different error rates are estimated for each locus
marker.type	"MSW" or "MSC" for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Kalinowski, 2006; Marshall, 1998) or "AFLP" for dominant markers (Hadfield, 2009).

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Marshall, T. C. *et al* (1998) Molecular Ecology 7 5 639-655 Kalinowski S.T. *et al* (2007) Molecular Ecology 16 5 1099-1106 Hadfield J. D. *et al* (2009) *in prep*

See Also

MCMCped

Examples

```
## Not run:
data(WarblerG)
GdP<-GdataPed(WarblerG)

## End(Not run)
```

---

genotype.list *Genotype Objects for all Loci*

Description

Creates a list of genotype objects from a matrix or data.frame of multilocus genotypes.

Usage

```
genotype.list(G, marker.type="MSW")
```

Arguments

G	matrix or data.frame of multilocus genotypes with individuals down the rows and loci across columns. Adjacent columns are taken to be the same locus
marker.type	"MSW" or "MSC" for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Kalinowski, 2006; Marshall, 1998) or "AFLP" for dominant markers (Hadfield, 2009).

Value

list of genotype objects for all loci

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Marshall, T. C. *et al* (1998) Molecular Ecology 7 5 639-655 Kalinowski S.T. *et al* (2007) Molecular Ecology 16 5 1099-1106 Hadfield J. D. *et al* (2009) *in prep*

See Also

genotype

Examples

```
## Not run:
  data(WarblerG)

  G<-genotype.list(WarblerG[,-1])
  summary(G[[1]])

## End(Not run)
```

---

genotypeD *genotypeD Object*

Description

Extends the genotype class for dominant marker data

Usage

```
genotypeD(a1, locus=NULL)
```

Arguments

a1	vector of scored genotypes (0 or 1) for dominant markers
locus	object of class locus, gene, or marker, holding information about the source of this genotype.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

genotype, summary.genotypeD

Examples

```
## Not run:
l1<-rbinom(100,1,0.5)
l1<-genotypeD(l1)

## End(Not run)
```

getXlist *Design Matrices for the Multinomial Log-Linear Model*

Description

Forms design matrices for each offspring, and stores other relevant information.

Usage

```
getXlist(PdP, GdP=NULL, A=NULL, E1=0.005, E2=0.005, mm.tol=999)
```

Arguments

PdP	PdataPed object
GdP	optional GdataPed object
A	optional list of allele frequencies. If not specified and GdP exists, allele frequencies are taken from GdP$G using extractA
E1	if Wang's (2004) model of genotyping error for co-dominant markers is used this is the probability of an allele dropping out. If CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers is used this parameter is not used. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a dominant allele being scored as a recessive allele.
E2	if Wang's (2004) or CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers are used this is the probability of an allele being miss-scored. In the CERVUS model errors are not independent for the two alleles within a genotype and so if a genotyping error has occurred at one allele then a genotyping error occurs at the other allele with probability one. Accordingly, E2(2-E2) is the per-genotype rate defined in CERVUS. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a recessive allele being scored as a dominant allele.
mm.tol	maximum number of genotype mismatches tolerated for potential parents

Details

This is the main R routine for setting up design matrices for the various models that may be defined in the formula argument of PdataPed. If a GdataPed object is passed to getXlist design matrices of genetic likelihoods are calculated (see fillX.G), and the number of mismatches between offspring and parental genotypes are stored (see mismatches). mm.tol specifies the maximum number of mismatches that are tolerated between an offspring and a parent. Parents that exceed this number of mismatches are excluded, and the design matrices for non-excluded parents are reordered by the number of mismatches. This increases the efficiency of sampling from the multinomial distribution of parents, because high probability parents appear first.

Value

id	vector of unique identifiers taken from PdP
beta_map	index relating the vector of unique parameters to the columns of the design matrices
X	list of design matrices and other information.

Note

Each element of X refers to an offspring (names(X)) and contains vectors for the set of potential parents (restdam.id and restsire.id) of each offspring. Also included are the set of individuals that may have been parents but have been excluded for certain reasons (dam.id and sire.id). Exclusion may have been based on the number of genotype mismatches, or it may have been on biological grounds (See the keep argument of varPed). Parental id's are stored as integers which correspond to the actual id's stored in id. Parental id's greater than the length of id refer to unsampled parents.

Six types of design matrix are used (XDus, XDs, XSus, XSs, XDSus, XDSs). XD.. are the design matrices for dams, and XS.. are the design matrices for sires. The rows of each design matrix are associated with individuals in dam.id and sire.id, respectively. When interactions between dam and sire variables are modelled, or a varPed variable is created using the argument relational="MATE", the design matrices vary over parental combinations. XDS.. are the design matrices for parental combinations with sire's varying the fastest. Each of these three types of design matrix have two subclasses: s and us. s are design matrices which are fully observed, either because unsampled parents do not exist or because unsampled parents have known phenotypes (see argument USvar in varPed). us are for design matrices where the phenotypes of unsampled parents are unknown. The matrices XDus and Xsus have a row of NA's which correspond to the unsampled parent category. The design matrix XDSus will typically have many rows of NA's because each sampled parent may be paired to an unsampled individual.

When the argument gender=NULL is passed to varPed the respective columns in the dam and sire design matrices are associated with a single parameter. Because of this the number of parameters to be estimated may be less than the total number of columns in the 6 design matrices. beta_map relates a parameter vector to the columns of the design matrices. The columns of the design matrices are numbered in the order they are introduced in the preceding paragraph (i.e XDus through to XDSs). The parameter vector is ordered identically except parameters associated with genderless variables are omitted for males. par_order is similar to beta_map but relates the order of the parameters specified in the formula argument to PdataPed to the respective columns of the design matrices.

If the argument relational="OFFSPRING" is specified in varPed, or the set of potential parents varies over offspring, the design matrices will vary across offspring. For this reason I create a design matrix for each offspring irrespective of whether the matrices vary or not. The design matrices for the genetic likelihoods will always vary over offspring.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31 Kalinowski S.T. *et al* (2006) Molecular Ecology *in press* Hadfield J. D. *et al* (2007) *in prep*

See Also

varPed, MCMCped

Examples

```
## Not run:
id<-1:20
sex<-sample(c("Male", "Female"),20, replace=TRUE)
offspring<-c(rep(0,18),1,1)
lat<-rnorm(20)
long<-rnorm(20)
mating_type<-gl(2,10, label=c("+", "-"))

test.data<-data.frame(id, offspring, lat, long, mating_type, sex)

res1<-expression(varPed("offspring", restrict=0))
var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))
var2<-expression(varPed(c("mating_type"), gender="Female",
  relational="MATE"))
var3<-expression(varPed("mating_type", gender="Male"))

PdP<-PdataPed(formula=list(res1, var1, var2, var3), data=test.data)

X.list<-getXlist(PdP)
X.list$X$"19"$XSs

# For the first offspring we have the design matrix for sires
# The first column represents the distance between each male
# and each offspring. The second column indicates the male's
# mating type. Note that contrasts are set up with the first
# male so the indicator variables may be negative.

matrix(X.list$X$"19"$XDSs, ncol=length(X.list$X$"19"$dam.id),
   nrow=length(X.list$X$"19"$sire.id))

# incidence matrix indicating whether Females (columns) and Males (rows)
# are the same mating type. Again this is a contrast with the first
# parental combination (which is +/+) so 0 actually represents parents
# with the same mating type.

## End(Not run)
```

---

insertPed *Inserts Founders into a Pedigree*

Description

Inserts Founders into a Pedigree

Usage

```
insertPed(ped, founders=NULL)
```

Arguments

ped	pedigree with id, dam and sire in ech column
founders	optional vector of founder id's. If not specified, then parents without their own pedigree row are inserted

Value

a pedigree pedigree with id, dam and sire in each column

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
pedigree<-matrix(NA, 7,3)
pedigree[,1]<-2:8
pedigree[,2][4:7]<-c(1,1,2,2)
pedigree[,3][4:7]<-c(3,3,4,4)

pedigree2<-insertPed(pedigree)

## End(Not run)
```

---

legalG *Legal Genotype Configurations*

Description

A function for checking whether a set of genotypes have a positive probability given the pedigree. If not, a legal configuration is found using heuristic methods. Missing genotypes are also replaced with compatible genotypes.

Usage

```
legalG(G, A, ped, time_born=NULL, marker.type="MSW")
```

Arguments

G	list of genotype objects
A	list of allele frequencies
ped	pedigree with id in the first column, dam in the second, and sire in the third. The genotypes must be in the same order as the id column
time_born	an optional vector for ordering a pedigree more efficiently (see orderPed)
marker.type	"MSW" or "MSC" for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Kalinowski, 2006; Marshall, 1998) or "AFLP" for dominant markers (Hadfield, 2009).

Value

G	a list of genotype objects with positive likelihood given the pedigree
legal	logical; TRUE if the the genotype configuration passed to legalG had a positive likelihood

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Marshall, T. C. *et al* (1998) Molecular Ecology 7 5 639-655 Kalinowski S.T. *et al* (2007) Molecular Ecology 16 5 1099-1106 Hadfield J. D. *et al* (2009) *in prep*

See Also

MCMCped

Examples

```
## Not run:
data(WarblerG)
A<-extractA(WarblerG[,16:17])

pedigree<-matrix(NA, 8,3)
pedigree[,1]<-1:8
pedigree[,2][5:8]<-c(1,1,2,2)
pedigree[,3][5:8]<-c(3,3,4,4)

G<-simgenotypes(A, E1=0, E2=0.3, ped=pedigree, no_dup=1)

newG<-legalG(G=G$Gobs,A=A,ped=pedigree)
newG$valid

# The input genotypes had a zero probability given the pedigree
# (because of genotype error) but the output genotypes have
# positive probability
```

```
legalG(newG$G,A,pedigree)$valid

## End(Not run)
```

`MasterBayes` *Maximum Likelihood and Markov chain Monte Carlo methods for Pedigree Reconstruction, Analysis and Simulation.*

Description

The primary aim of MasterBayes is to use MCMC techniques to integrate over uncertainty in pedigree configurations estimated from molecular markers and phenotypic data. Emphasis is put on the marginal distribution of parameters that relate the phenotypic data to the pedigree. All simulation is done in compiled C++ using the Scythe Statistical Library. More detailed information can be found in `vignette("Tutorial", "MasterBayes")`.

Details

The motivation behind the package is to approximate the following probability distribution using Markov chain Monte Carlo techniques:

$$p(\beta|\mathbf{G},\mathbf{y})$$

where β is the vector of parameters of primary interest, $\mathbf{G}$ are the genetic data and $\mathbf{y}$ are phenotypic data. Generally, it is not possible to simulate from the posterior distribution of β when the problem is in this form and so I augment the parameter space with the pedigree, $\mathbf{P}$:

$$\int_{\mathbf{P}} p(\beta,\mathbf{P}|\mathbf{G},\mathbf{y})\mathbf{dP}$$

This simplifies the problem because the likelihood can be expressed more simply:

$$L(\mathbf{G},\mathbf{y}|\beta,\mathbf{P}) = \mathbf{L}(\mathbf{G}|\mathbf{P})\mathbf{L}(\mathbf{y}|\mathbf{P},\beta)$$

This simplification rests on the assumption that the genetic and non-genetic data are independent after conditioning on the pedigree. This will generally be true when markers are not linked to QTL's. The first likelihood, $L(\mathbf{G}|\mathbf{P})$, is easily calculated for arbitrary pedigrees using the Elston-Stewart algorithm (Elston, 1971), and is based around the Mendelian transition probability. The second likelihood is obtained by fitting the multinomial log-linear model:

$$L(\mathbf{y}|\beta,\mathbf{P}) \propto \mathbf{p}(\mathbf{P}|\beta,\mathbf{y})\mathbf{p}(\mathbf{P}).$$

Assuming that the set of possible pedigrees have equal prior probability, and that offspring are independently distributed after conditioning on the predictor variables:

$$L(\mathbf{y}|\beta,\mathbf{P}) \propto \prod_{\mathbf{i=1}}^{\mathbf{n_o}} \frac{\mathbf{e}^{\mathbf{X}^{\mathbf{i}}_{\mathbf{p_i}}\beta}}{\sum_{\mathbf{j=1}}^{\mathbf{n_p}} \mathbf{e}^{\mathbf{X}^{\mathbf{i}}_{\mathbf{j}}\beta}}.$$

where $\mathbf{X}^{\mathbf{i}}_{\mathbf{j}}$ denotes the j^{th} row of offspring i's design matrix formed from the phenotypic data $\mathbf{y}$. Each row of the design matrix corresponds to a parental combination. n_o and n_p denote the number of offspring and the number of potential parental combinations, respectively. p_i denotes the actual parents of individual i (Smouse, 1999).

This likelihood is evaluated over the probability distribution of the pedigree, $\mathbf{P}$:

$$p(\mathbf{P}|\mathbf{G},\mathbf{y},\beta).$$

Most other techniques approximate this distribution as $p(\mathbf{P}|\mathbf{G})$, and even then tend to use the mode rather than the complete distribution, leading to inferential problems (See the information boxes in Hadfield et al. 2006).

Unfortunately, genotype data are rarely observed with out error and the parents of some offspring may not be sampled. If the genetic markers are co-dominant then genotyping errors can be handled following either the model of Wang (2004) or CERVUS (Marshall 1998). When the markers are dominant I model the probabilities of a dominant allele being miss-scored as a recessive and *vice versa*. Denoting the parameters associated with these two forms of genotyping error as ε_1 and ε_2, and the vector of parental allele frequencies as ω, two solutions are implemented.

An exact solution:

$$\int_{\mathbf{P}}\int_{\mathbf{G}}\int_{\varepsilon_1}\int_{\varepsilon_2}\int_{\omega} p(\beta,\mathbf{P},\mathbf{G},\varepsilon_{\mathbf{1}},\varepsilon_{\mathbf{2}}.\omega|\mathbf{G}^{(\mathbf{obs})},\mathbf{y})\mathbf{d}\mathbf{P}\mathbf{d}\mathbf{G}\mathbf{d}\varepsilon_{\mathbf{1}}\mathbf{d}\varepsilon_{\mathbf{2}}\mathbf{d}\omega$$

where the posterior probability distribution of the error rates, the allele frequencies and the true unobserved genotypes, $\mathbf{G}$, are estimated and integrated out. The conditional distribution of the true genotypes in the exact form is given by:

$$p(\mathbf{G^{obs}}|\mathbf{G},\varepsilon_{\mathbf{1}},\varepsilon_{\mathbf{2}})\mathbf{p}(\mathbf{G}|\mathbf{P},\omega)$$

The second solution is an approximation to the above equation, and uses point estimates for ω, ε_1 and ε_2. The conditional distribution of $\mathbf{G}$ is derived ignoring the information present in $\mathbf{P}$:

$$p(\mathbf{G^{obs}}|\mathbf{G},\varepsilon_{\mathbf{1}},\varepsilon_{\mathbf{2}})\mathbf{p}(\mathbf{G}|\omega)$$

The approximation can be derived analytically, whereas the exact solution requires the Markov chain to be augmented with the true genotypes of all individuals. This becomes very computer intensive but the approximation breaks down for dominant markers, or models in which the number of unsampled males and/or females is to be estimated. Unsampled parents are dealt with, and their number estimated using an approximation originally due to Nielsen (2001). An exact solution to the problem has been proposed by Emery *et.al.* (2001) but becomes impractical as the number of unsampled parents gets large. Nielsen's approximation is based around the Mendelian transition probability when a parental genotype is unknown. This probability is derived using estimates of the allele frequencies at that locus and the assumption of Hardy-Weinberg equilibrium.

I deal with the fact that unsampled individuals have missing phenotype data by approximating the distribution of the sum of linear predictors across unsampled parents. This approximation relies on the assumption that the unsampled individuals come from the same statistical population as sampled individuals, and that population sizes are large enough so that the distribution for the sum tends to a normal distribution under the central limit theorem.

Taking n and N as the number of sampled individuals, and the total number of individuals in the population respectively:

$$p(\sum^{N-n} \hat{\mathbf{p}}^{(\mathbf{miss})}|\hat{\mathbf{p}}^{(\mathbf{obs})}) \approx \mathbf{N}(\frac{\mathbf{N}-\mathbf{n}}{\mathbf{n}}\sum^{\mathbf{n}} \hat{\mathbf{p}}^{(\mathbf{obs})}, \frac{\mathbf{N}(\mathbf{N}-\mathbf{n})}{\mathbf{n}}\mathbf{S}^{\mathbf{2}}_{\mathbf{obs}})$$

where $\hat{\mathbf{p}}$ are vectors of linear predictors for the unsampled $^{(miss)}$ and sampled $^{(obs)}$ individuals, respectively (Gelman *et al.*, 2004). S^2_{obs} is the sample variance of the observed linear predictors.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Elston, R. C. and Stewart, J. Human Heredity (1971) 21 523-542 Emery, A. M. *et.al* Molecular Ecology (2001) 10 1265-1278 Gelman, A. *et.al* Bayesian Data Analysis *Edition II* (2004) Chapman and Hall Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31 Marshall, T. C. *et al* (1998) Molecular Ecology 7 5 639-655 Nielsen. R. *et.al* Genetics (2001) 157 4 1673-1682 Smouse P.E. *et al* (1999) Journal of Evolutionary Biology 12 1069-1077 Wang J.L. Genetics (2004) 166 4 1963-1979

See Also

MCMCped

MCMCped *Markov chain Monte Carlo Methods for Pedigree Reconstruction and Analysis*

Description

Markov chain Monte Carlo methods for estimating the joint posterior distribution of a pedigree and the parameters that predict its structure using genetic and non-genetic data. These parameters can be associated with covariates of fecundity such as a sexually selected trait or age, or can be associated with spatial or heritable traits that relate parents to specific offspring. Population size, allele frequencies, allelic dropout rates, and stochastic genotyping error rates can also be simultaneously estimated.

Usage

```
MCMCped(PdP=PdataPed(), GdP=GdataPed(), sP=startPed(), tP=tunePed(),
   pP=priorPed(), mm.tol=999, nitt = 13000, thin = 10,  burnin =
   3000, write_postG = FALSE, write_postA=FALSE, write_postP =
   "MARGINAL", checkP = FALSE, jointP = TRUE, DSapprox=FALSE, verbose=TRUE)
```

Arguments

PdP	optional PdataPed object containing phenotypic data
GdP	optional GdataPed object containing genetic data
sP	optional startPed object containing starting parameterisation
tP	optional tunePed object containg tuning parameters for Metropolis Hastings updates
pP	optional priorPed object containg prior specifications
mm.tol	maximum number of mismatches tollerated
nitt	number of MCMC iterations
thin	thinning interval of the Markov chain
burnin	the number of initial iterations to be discarded
write_postG	if TRUE the marignal posterior distribution of true genotypes is stored
write_postA	if TRUE the joint posterior distribution of allele frequencies is stored
write_postP	if "MARGINAL" the marginal distribution of parents is stored. If "JOINT" the joint distribution of parents (the pedigree) is stored.
checkP	if TRUE the pedigree is checked for legality, and illegal pedigrees rejected. If FALSE it is assumed that any potential parent would produce a legal pedigree, i.e one without circuits, in the terminology of graph theory.
jointP	if TRUE both parents are sampled simultaneously, if FALSE each parent is sampled conditional on the other. TRUE should mix faster, but FALSE should iterate faster, especially when relational="MATE" is passed to varPed
DSapprox	if TRUE the likelihood for models in which a relational="MATE" variable is passed is approximated. This can be much more efficient because the denominator of the multinomial is the summed linear pedictors for combinations in which i=m or j=m where m referes to the "MATE" at the current iteration.
verbose	if TRUE posterior samples and the Metropolis Hastings accpetance rates of beta, USdam, USsire, E1, E2 are printed to the screen every 1000 iterations.

Value

beta	an mcmc object containing samples from the posterior distribution of the population level parameters
USdam	an mcmc object containing samples from the posterior distribution of the number of unsampled females
USsire	an mcmc object containing samples from the posterior distribution of the number of unsampled males

E1	an mcmc object containing samples from the posterior distribution of allelic dropout rates for codominant markers or the probability of mis-scoring a dominant allele as recessive for dominant markers
E2	an mcmc object containing samples from the posterior distribution of stochasting genotyping error rates for codominant markers or the probability of mis-scoring a recessive allele as dominant for dominant markers
G	list of marginal distributions of true genotypes at each locus
A	list of mcmc objects containing samples from the posterior distribution of the base population allele frequencies at each locus
P	either samples from the posterior distribution of the pedigree, or the marginal distribution of parents

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31

See Also

getXlist

Examples

```
data(WarblerP)
data(WarblerG)

GdP<-GdataPed(WarblerG)

var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))

# paternity is to be modelled as a function of distance
# between offspring and male territories

res1<-expression(varPed("offspring", restrict=0))

# indivdiuals from the offspring generation are excluded as parents

res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

# mothers not from the offspring territory are excluded

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
tP<-tunePed(beta=30)
```

```
model1<-MCMCped(PdP=PdP, GdP=GdP, tP=tP, nitt=300, thin=1, burnin=0)

plot(model1$beta)
```

`mismatches` *Parent-Offspring Genotype Mismatches*

Description

Calculates the number of mismatches between parental and offspring genotypes, assuming the genotypes of spouses are unknown. Primarily intended to be used inside the function `getXlist` where potential parents can be excluded based on the number of mismatches. Dominant markers do not produce mismatches.

Usage

```
mismatches(X.list, G, mm.tol=999)
```

Arguments

`X.list`	list of design matrices for each offspring derived using `getXlist`
`G`	list of genotype objects, the rows of which must refer to the id vector `X.list$id`
`mm.tol`	maximum number of mismatches that are tolerated before exclusion

Value

list of design matrices of the form `X.list`, but containing the number of mismatches between parents and offspring. Potential parents that exceed the number of mismatches specified by `mm.tol` are removed from the vectors of potential parents: `restdam.id` and `restsire.id`.

Note

If a `GdataPed` object is passed to `getXlist` then the number of mismatches will be calculated by default.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

`MCMCped`

Examples

```
## Not run:
data(WarblerG)
A<-extractA(WarblerG)

ped<-matrix(NA, 5,3)
ped[,1]<-1:5
ped[,2]<-c(rep(NA, 4), 1)
ped[,3]<-c(rep(NA, 4), 2)

genotypes<-simgenotypes(A, ped=ped)

sex<-c("Female", "Male", "Female", "Male","Female")
offspring<-c(0,0,0,0,1)

data<-data.frame(id=ped[,1], sex, offspring)

res1<-expression(varPed(x="offspring", restrict=0))
PdP<-PdataPed(formula=list(res1), data=data)

X.list<-getXlist(PdP)
# creates design matrices for offspring (in this case indivdiual "5")

X.list.MM<-mismatches(X.list, G=genotypes$Gobs, mm.tol=0)
# genetic likelihoods are arranged sires within dams

X.list.MM$X$"5"$mmD
# number of mismatches between offspring "5" and dams "1" and "3"

X.list.MM$X$"5"$mmS
# number of mismatches between offspring "5" and sires "4" and "5"

X.list.MM$X$"5"$restdam.id
X.list.MM$X$"5"$dam.id
# dams with mismatches are excluded  mismatch (mm.tol=0)

X.list.MM$X$"5"$restsire.id
X.list.MM$X$"5"$sire.id
# sires with mismatches are excluded  mismatch (mm.tol=0)

## End(Not run)
```

MLE.beta *Maximum Likelihood Estimation of Beta*

Description

Finds MLE for beta given a pedigree, via a call to optim. Beta is the paramater vector of a multinomial log-linear model.

Usage

```
MLE.beta(X.list, ped, beta=NULL, nUSdam=NULL, nUSsire=NULL, shrink=NULL)
```

Arguments

X.list	list of design matrices for each offspring derived using getXlist
ped	pedigree with id, dam and sire in ech column
beta	optional starting vector for beta
nUSdam	optional number of unsampled females. Only required if unsampled females have known phenotype.
nUSsire	optional number of unsampled males. Only required if unsampled males have known phenotype.
shrink	optional scalar for the variance defining the ridge-regression likelihood penalisation.

Value

beta	vector of MLE's for beta
C	large sample variance-covariance matrix of beta MLE's

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31 Smouse P.E. *et al* (1999) Journal of Evolutionary Biology 12 1069-1077

See Also

MCMCped, beta.loglik

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)

GdP<-GdataPed(WarblerG)

res1<-expression(varPed("offspring", restrict=0))
var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))
res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
```

```
X.list<-getXlist(PdP=PdP, GdP=GdP, E2=0.005)

ped<-MLE.ped(X.list)$P
beta<-MLE.beta(X.list, ped)
beta

## End(Not run)
```

MLE.ped *Maximum Likelihood Estimation of the Pedigree*

Description

Finds the MLE pedigree using the genetic data only. An approximation is used for genotyping error.

Usage

```
MLE.ped(X.list, ped=NULL, USdam=FALSE, nUSdam=NULL, USsire=FALSE,
   nUSsire=NULL, threshold=0, checkP)
```

Arguments

X.list	list of design matrices for each offspring derived using getXlist
ped	optional pedigree with id, dam and sire in ech column
USdam	logical or character; if TRUE a single undiferentiated population of unsampled females exists. If USdam is a character vector it must have the same length as id with factor levels representing sub-populations (in time or space) over which the number of unsampled females vary.
nUSdam	numeric vector for number of unsampled females
USsire	logical or character; if TRUE a single undiferentiated population of unsampled males exists. If USsire is a character vector it must have the same length as id with factor levels representing sub-populations (in time or space) over which the number of unsampled males vary.
nUSsire	numeric vector for number of unsampled males
threshold	threshold probability under which ML parents are replaced by NA
checkP	if TRUE the pedigree is checked for legality, and illegal pedigrees rejected. If FALSE it is assumed that any potential parent would produce a legal pedigree, i.e one without circuits, in the terminology of graph theory. Legality is checked

Details

ML estimation of the pedigree is based on the Mendelian transition probabilities in the presence of genotyping error as outlined in Kalinwoski (2006). The probability that the ML parents are the true parents is simply the Mendelian transition probability for those parents divided by the sum of the transition probabilities for the remaining potential parents, both sampled and unsampled. If ped exists and the dam column contains known dam assignemnts and the sire column contains only

NA's, then the ML sires will be returned conditional on the dam assignements being true. ML dam estimation with known sires can be performed in the same way. Individuals whose parents cannot be assigned with the required level of certainty (`threshold`), or whose parents belong to the base or unsampled population, have NA in the dam and sire columns. If each indiviual's potential parents are such that an illegal pedigree could be sampled then `checkP=TRUE` can be used to ensure legality. This is recommended if the pedigree is to be passed as a starting pedigree to MCMCped. It should be noted that under these circumstances it is possible that multiple pedigrees max exist with the same likelihood and this may not be obvious from the MLE.ped output since assignments are made conditional on earlier assignement being true. As an example, if there are two indiviuals both of which could potentially be each others parents then assigning both to be each others parent is illegal (since each indiviual would be its own grandparent). In simple situations, the parent-offspring and offspring-parent assignements have equal probability, but when `checkP=TRUE` the first indiviual would have zero probability of being the second individual's parent if the second individual was already assigned as the first individual's parent.

Value

`P`	pedigree with id in the first column, and dam and sire in the second and third columns
`prob`	probability of the most likely parental combination

Author(s)

Jarrod Hadfield `<j.hadfield@ed.ac.uk>`

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31 Marshall J.D. *et al* (1998) Molecular Ecology 7 639-655 Kalinowski S.T. *et al*, Molecular Ecology *in press*

See Also

`MCMCped`

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)

GdP<-GdataPed(WarblerG)

res1<-expression(varPed("offspring", restrict=0))
res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

PdP<-PdataPed(formula=list(res1,res2), data=WarblerP, USsire=TRUE)

X.list<-getXlist(PdP=PdP, GdP=GdP, E2=0.005)
```

```
ped<-MLE.ped(X.list, USsire=TRUE, nUSsire=10, threshold=0.75)

## End(Not run)
```

`MLE.popsize` *Maximum Likelihood Estimation of the Unsampled Population Size*

Description

Finds the MLE for the number of unsampled males and/or females following Nielsen *et al.* (2001). The size of the unsampled population can vary over time and space, and genotyping error is accomodated using the CERVUS model of genotyping error (Kalinwoski *et al.* 2006).

Usage

```
MLE.popsize(X.list, USdam=FALSE, USsire=FALSE, nUS=NULL,
   ped=NULL, shrink=NULL)
```

Arguments

`X.list`	list of design matrices for each offspring derived using `getXlist`
`USdam`	logical or character; if `TRUE` a single undiferentiated population of unsampled females exists. If `USdam` is a character vector it must have the same length as the number of offspring (`length(X.list$X)`) with factor levels representing sub-populations (in time or space) over which the number of unsampled females vary.
`USsire`	logical or character; if `TRUE` a single undiferentiated population of unsampled males exists. if `USsire` is a character vector it must either have the same length as the number of offspring (`length(X.list$X)`) with factor levels representing sub-populations (in time or space) over which the number of unsampled males vary, or alternatively `"USdam"`, in which case the unsampled male and female populations are constrained to be equal.
`nUS`	optional starting vector for the size of the unsampled population. Parmeters for the unsampled female population come before the male population.
`ped`	optional pedigree with id, dam and sire in ech column
`shrink`	optional scalar for the variance defining the ridge-regression likelihood penalisation.

Value

`nUS`	vector of MLE's for the size of the unsampled population. Lower bound is 1e-5 for numerical stability.
`C`	large sample variance-covariance matrix of `nUS` MLE's

Note

Nielsen's original model does not account for genotyping error, and estimation of the unsampled population size is VERY sensitive to the level of genotyping error. This function implements a commonly used approxiamtion for genotyping error that ignores pedigree information. For many problems this approximation seems valid, but appears to break down when estimating the size of the unsampled population size. Bayesian estimation of the unsampled population size (see MCMCped) that uses an exact solution for genotyping error is more robust.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Nielsen. R. *et.al* Genetics (2001) 157 4 1673-1682

See Also

MCMCped, popsize.loglik

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)

GdP<-GdataPed(WarblerG)
res1<-expression(varPed("offspring", restrict=0))

PdP<-PdataPed(formula=list(res1), data=WarblerP, USsire=TRUE, USdam=TRUE)

X.list<-getXlist(PdP=PdP, GdP=GdP, E2=0.02)

nUS<-MLE.popsize(X.list, USsire=TRUE, USdam=TRUE)
nUS

## End(Not run)
```

---

modeG *Posterior Mode of Genotypes*

Description

Finds the mode of the posterior marginal distribution of genotypes

Usage

```
modeG(postG, threshold=0)
```

Arguments

postG	posterior distribution of genotypes from an MCMCped model with argument write_postG=TRUE
threshold	threshold probability under which ML genotypes are replaced by NA

Value

G	list of genotype objects
id	id vector

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al*, Molecular Ecology

See Also

MCMCped, genotype

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)
GdP<-GdataPed(WarblerG)

var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))

# paternity is to be modelled as a function of distance
# between offspring and male territories

res1<-expression(varPed("offspring", restrict=0))

# indivdiuals from the offspring generation are excluded as parents

res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

# mothers not from the offspring territory are excluded

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
tP<-tunePed(beta=30)

model1<-MCMCped(PdP=PdP, GdP=GdP, tP=tP, nitt=3000, thin=2, burnin=1000, write_postG=TRUE)

G<-modeG(model1$G)$G
summary(G[[1]])
```

```
## End(Not run)
```

---

modeP *Posterior Mode of Parents*

Description

Finds the mode of the posterior marginal distribution of parents

Usage

```
modeP(postP, threshold=0, marginal=FALSE, USasNA=TRUE)
```

Arguments

postP	posterior distribution of parentage
threshold	threshold probability under which ML parents are replaced by NA
marginal	logical; should the marginal mode be calculated from the joint distribution?
USasNA	logical; should usampled parents be replaced by NA?

Details

Individuals that do not have a parent assignment with a posterior probability exceeding the threshold, or whose parents belong to the base or unsampled population (if USasNA=TRUE), have NA as their parents. Please bear in mind that the mode of the marginal distribution (returned by MCMCped if write_postP="MARGINAL") may be different from the mode of the joint distribution (write_postP="JOINT"). For example the male that has the highest marginal probability (marginal with respect to potential mothers) may not be the male that is in the parental category (i.e. dam/sire combination) with the highest probability. If write_postP="JOINT" was sepcified, then the mode of the marginal distribution can be obtained by specifying marginal=TRUE. The modes are marginal with respect to other offspring and with multigenerational pedigrees may not coincide with the mode of the distribution of pedigrees.

Value

P	pedigree with id in the first column, and dam and sire in the second and third columns
prob	marginal posterior probability of the most likely parental combination (joint) or the most likely mother (marginal)
prob.male	marginal posterior probability of the most likely father (marginal)

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)
GdP<-GdataPed(WarblerG)

var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))

# paternity is to be modelled as a function of distance
# between offspring and male territories

res1<-expression(varPed("offspring", restrict=0))

# indivdiuals from the offspring generation are excluded as parents

res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

# mothers not from the offspring territory are excluded

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
tP<-tunePed(beta=30)

model1<-MCMCped(PdP=PdP, GdP=GdP, tP=tP, nitt=3000, thin=2, burnin=1000)

ped<-modeP(model1$P, threshol=0.9)
ped

## End(Not run)
```

---

orderPed *Orders a Pedigree*

Description

Orders a pedigree so parents come before offspring

Usage

```
orderPed(ped, time_born=NULL)
```

Arguments

ped	pedigree with id, dam and sire in ech column
time_born	an optional vector of birth dates by which the pedigree can be ordered)

Value

an ordered pedigree pedigree with id, dam and sire in each column

Note

This function has changed name from `order.ped` in earler versions <2.42. `order.ped` did not always (rarely) ordered the pedigree correctly. This new function uses the `kindepth` function from the `kinship2` package

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
pedigree<-matrix(NA, 8,3)
pedigree[,1]<-1:8
pedigree[,2][5:8]<-c(1,1,2,2)
pedigree[,3][5:8]<-c(3,3,4,4)

pedigree<-pedigree[sample(1:8),]

pedigree2<-orderPed(pedigree)

## End(Not run)
```

PdataPed *PdataPed Object*

Description

`PdataPed` creates an object of class `PdataPed`, which typically contains the phenotype data to be passed to MCMCped and the formula that defines the model to be fitted. `is.PdataPed` returns `TRUE` if `x` is of class `PdataPed`

Usage

```
PdataPed(formula, data=NULL, id=data$id, sex=data$sex,
   offspring=data$offspring, timevar=data$timevar,
   USdam=FALSE, USsire=FALSE)
```

Arguments

formula	list of model predictors of the form expression(varPed(...))
data	data frame containing the predictor variables
id	vector of individual identifiers. If not specified, data must have an id column
sex	vector of individual sexes (either 'Male' or 'Female' or NA). If not specified individuals are assumed to be hermpahroditic unless data has a sex column
offspring	binary vector indicating whether records belong to offspring (1) or not (0)
timevar	an optional vector indicating cohorts for multigenerational pedigree reconstruction
USdam	logical or character; if TRUE a single undiferentaited population of unsampled females exists. If USdam is a character vector it must have the same length as id with factor levels representing sub-populations (in time or space) over which the number of unsampled females vary.
USsire	logical or character; if TRUE a single undiferentaited population of unsampled males exists. If USsire is a character vector it must have the same length as id with factor levels representing sub-populations (in time or space) over which the number of unsampled males vary.

Details

If the number of unsampled individuals varies over subpopulations, and the parentage of an offspring is not restricted to ceratin subpopulations then the parameters will not be idenifiable. This can be resolved by using an informative prior (see priorPed) for the number of unsampled individuals in each sub-population, or using the restrict argument in varPed.

Value

list containing the arguments passed

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
id<-1:20
sex<-sample(c("Male", "Female"),20, replace=TRUE)
offspring<-c(rep(0,18),1,1)
lat<-rnorm(20)
long<-rnorm(20)
mating_type<-gl(2,10, label=c("+", "-"))

test.data<-data.frame(id, offspring, lat, long, mating_type, sex)
```

```
res1<-expression(varPed("offspring", restrict=0))
var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))
var2<-expression(varPed(c("mating_type"), gender="Female",
  relational="MATE"))
var3<-expression(varPed("mating_type", gender="Male"))

PdP<-PdataPed(formula=list(res1, var1, var2, var3), data=test.data)
```

`popsize.loglik` *Log-Likelihood of Unsampled Population Size*

Description

Log-likelihood of the number of unsampled individuals given the genotypes of offspring and potential parents

Usage

```
popsize.loglik(X, USdam=FALSE, USsire=FALSE, nUS=NULL, ped=NULL, USsiredam=FALSE,
   shrink=NULL)
```

Arguments

`X`	list for each offspring with elements `N` and `G`. `N` is a vector conatining the number of parental combinations in each of 4 classes. `G` is a vector conatining the sum of the Mendelian transition probabilities over parental combinations in each class. The 4 classes are parental combinations where a) both parents are sampled b) only sires are sampled, c) only dams are sampled d) neither parent is sampled.
`USdam`	logical or character; if `TRUE` a single undiferentiated population of unsampled females exists. if `USdam` is a character vector it must have the same length as `id` with factor levels representing sub-populations (in time or space) over which the number of unsampled females vary.
`USsire`	logical or character; if `TRUE` a single undiferentiated population of unsampled males exists. if `USsire` is a character vector it must have the same length as `id` with factor levels representing sub-populations (in time or space) over which the number of unsampled males vary.
`nUS`	vector for the size of the unsampled populations. Parmeters for the unsampled female populations come before the male populations.
`ped`	optional pedigree with id, dam and sire in ech column
`USsiredam`	logical; if `TRUE` male and female unsampled populations sizes are constrained to be equal
`shrink`	optional scalar for the variance defining the ridge-regression likelihood penalisation.

Value

log-likelihood of the number of unsampled individuals given the genotype data.

Note

Intended to be used within MLE.popsize

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Nielsen. R. *et.al* Genetics (2001) 157 4 1673-1682

See Also

MCMCped, MLE.popsize

Examples

```
## Not run:
data(WarblerG)
A<-extractA(WarblerG)

sex<-c(rep("Male", 50), rep("Female", 100))
offspring<-c(rep(0,100), rep(1, 50))
terr<-as.factor(rep(1:50, 3))
id<-1:150

res1<-expression(varPed(x="offspring", restrict=0))
res2<-expression(varPed(x="terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

test.data<-data.frame(id, sex, offspring, terr)

PdP<-PdataPed(formula=list(res1, res2), data=test.data)

simped<-simpedigree(PdP)
G<-simgenotypes(A, E1=0, E2=0, ped=simped$ped, no_dup=1)

# remove 25 males at random, leaving 25

rm.males<-sample(1:50, 25, replace=FALSE)

data.rm<-test.data[-rm.males,]
GdPrm<-GdataPed(G=lapply(G$Gobs, function(x){x[-rm.males]}),
  id=G$id[-rm.males])

# delete genotype and phenotype records

PdPrm<-PdataPed(formula=list(res1, res2), data=data.rm, USsire=TRUE)
```

```
X.listrm<-getXlist(PdP=PdPrm, GdP=GdPrm, A=A, E2=0)

X<-lapply(X.listrm$X, function(x){list(N=c(25,0,1,0),
  G=c(sum(x$G[1:25]), 0, x$G[26], 0))})

# each offspring has 1 mother and 25 sampled fathers so the 4 classes are:
# a) 1*25 categories with both parents sampled, 0*25 categries with only
# sires sampled b) 1*1 categories with only dams sired and 0*0 categories
# with both sexes unsampled.

nUS<-seq(10,40, length=100)
nUS_Loglik<-1:100
for(i in 1:100){
 nUS_Loglik[i]<-popsize.loglik(X, USsire=TRUE, nUS=nUS[i])
}
plot(nUS_Loglik~nUS, type="l", main="Profile Log-likelihood
   for number of unsampled males")

## End(Not run)
```

`post.pairs` *Returns pairs of individuals that fall into specific relatedness categories*

Description

Computes posterior probabilities of pairs of indiviuals falling into specific relatedness categories (parent-offsping, sibs, full-sibs, half-sibs). Returns those pairs that have a posterior probability greater than some threshold.

Usage

```
post.pairs(postP, threshold=0, rel="PO")
```

Arguments

`postP`	joint posterior distribution of parentage
`threshold`	threshold probability over which related pairs are returned
`rel`	relatedness category. Currently `"PO"` (Parent-Offspring), `"S"` (Sibs), `"FS"` (Full-Sibs) and `"HS"` (Half-Sibs) are supported.

Value

`P`	pairs of indiviuals that fall into the `rel` category with posterior probability > `threshold`
`prob`	posterior probability

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

modeP

Examples

```
## Not run:
data(WarblerP)
data(WarblerG)
GdP<-GdataPed(WarblerG)

var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))

# paternity is to be modelled as a function of distance
# between offspring and male territories

res1<-expression(varPed("offspring", restrict=0))

# indivdiuals from the offspring generation are excluded as parents

res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

# mothers not from the offspring territory are excluded

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
tP<-tunePed(beta=30)

model1<-MCMCped(PdP=PdP, GdP=GdP, tP=tP, nitt=3000, thin=2, burnin=1000, write_postP="JOINT")

fsib<-post.pairs(model1$P, threshol=0.9, rel="FS")
fsib$P

## End(Not run)
```

---

priorPed *priorPed Object*

Description

An object containing the prior specifiactions for a model fitted using MCMCped. If prior distributions are not specified then improper priors are used, and a proper posterior distribution cannot be gauranteed.

Usage

```
priorPed(E1=999, E2=999, beta=list(mu=999, sigma=999),
         USdam=list(mu=999, sigma=999),
         USsire=list(mu=999, sigma=999))
```

Arguments

E1	matrix of parameters for the beta distribution specifying the prior distribution. If Wang's (2004) model of genotyping error for co-dominant markers is used this is the probability of an allele dropping out. If CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers is used this parameter is not used. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a dominant allele being scored as a recessive allele. Rows correspond to error rate categories, columns to the beta shape parameters. The order of rows in E1 are the order in which the error rate categories appear in the categories argument of GdataPed (see dbeta). If perlocus=TRUE was passed to GdataPed, then the error rate categories are replicated across loci
E2	matrix of parameters for the beta distribution specifying the prior distribution. If Wang's (2004) or CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers are used this is the probability of an allele being miss-scored. In the CERVUS model errors are not independent for the two alleles within a genotype and so if a genotyping error has occurred at one allele then a genotyping error occurs at the other allele with probability one. Accordingly, E2(2-E2) is the per-genotype rate defined in CERVUS. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a recessive allele being scored as a dominant allele. Rows correspond to error rate categories, columns to the beta shape parameters. The order of rows in E1 are the order in which the error rate categories appear in the categories argument of GdataPed (see dbeta). If perlocus=TRUE was passed to GdataPed, then the error rate categories are replicated across loci
beta	list containing a vector for the mean, and a matrix for the variance-covariances of a multivariate normal distribution, that specifies the prior distribution for the population level parameters. The order of beta is the order in which the parameters appear in the MCMC ouput.
USdam	list containing vectors of means and standard deviations for log normal distributions that specify the prior distribution for the number of unsampled females. The order of USdam is the order in which the unsampled dam populations appear in the USdam argument of PdataPed (see dlnorm)
USsire	list containing vectors of means and standard deviations for log normal distributions that specify the prior distribution for the number of unsampled males. The order of USsire is the order in which the unsampled sire populations appear in the USsire argument of PdataPed (see dlnorm)

Value

list containing the arguments passed

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
# When each individual has only been genotyped once, and no pedigree
# information exists, there is virtually no information available
# to estimate error rates.  The tiny amount of information comes
# (dangerously) from the assumption of Hardy-Weinburg equilibrium.
# The posterior distribution is similar to the prior:

data(WarblerG)
A<-extractA(WarblerG)

ped<-matrix(NA, 100,3)
ped[,1]<-1:100

G<-simgenotypes(A, E1=0.01, E2=0.01, ped=ped, no_dup=1)
GdP<-GdataPed(G=G$Gobs, id=G$id)
pP<-priorPed(E1=matrix(c(40,1600), nrow=1), E2=matrix(c(40,1600), nrow=1))

model1<-MCMCped(GdP=GdP, pP=pP)

#The posterior distribution recovers the prior distribution

summary(model1$E1)
quantile(rbeta(1000, 40, 1600), prob=c(0.025, 0.25, 0.5, 0.75, 0.975))

## End(Not run)
```

---

reordXlist *Reorders Design Matrices*

Description

Reorders design matrices so excluded parents appear last, and high probability parents appear first, thus increasing computational efficiency.

Usage

```
reordXlist(X.list, marker.type="MSW")
```

Arguments

X.list	list of design matrices for each offspring derived using getXlist. Mismatch information must be present (see mismatches)
marker.type	"MSW" or "MSC" for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Kalinowski, 2006; Marshall, 1998) or "AFLP" for dominant markers (Hadfield, 2009).

Details

The design matrices are reordered by the number of mismatches between a parent and offspring for codominant markers, and by the probability of the offspring genotype conditional on parent genotype for dominant markers.

Value

X.list for which parents are reordered

Note

If a GdataPed object is passed to getXlist then the design matrices will be reordered by default.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
data(WarblerG)
A<-extractA(WarblerG)

ped<-matrix(NA, 5,3)
ped[,1]<-1:5
ped[,2]<-c(rep(NA, 4), 3)
ped[,3]<-c(rep(NA, 4), 4)

genotypes<-simgenotypes(A, ped=ped)

sex<-c("Female", "Male", "Female", "Male","Female")
offspring<-c(0,0,0,0,1)

data<-data.frame(id=ped[,1], sex, offspring)

var1<-expression(varPed(x="offspring", restrict=0))
PdP<-PdataPed(formula=list(var1), data=data)

X.list<-getXlist(PdP)
```

```
# creates design matrices for offspring (in this case indivdiual "5")

X.list<-mismatches(X.list, G=genotypes$Gobs)
X.list<-fillX.G(X.list, A=A, G=genotypes$Gobs)

X.list.reord<-reordXlist(X.list)

# The design matrices for the genetic likelihoods are reordered
# by the number of mismatches.  The true parental combination
# now appears first rather than last.

X.list$X$"5"$G
X.list.reord$X$"5"$G

## End(Not run)
```

`simgenotypes` *Genotype and Genotyping Error Simulation*

Description

Simulates genotypes given a pedigree and allele frequencies. Option exists to simulate observed genotypes given Wangs's (2004) or CERVUS's model (Marshall 1998) of genotyping error for codominat markers or an asymmetric allele based model for dominant markers (Hadfield, 2009).

Usage

```
simgenotypes(A, E1 = 0, E2 = 0, ped, no_dup = 1, prop.missing=0, marker.type="MSW")
```

Arguments

`A`	list of allele frequencies at each locus
`E1`	if Wang's (2004) model of genotyping error for co-dominant markers is used this is the probability of an allele dropping out. If CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers is used this parameter is not used. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a dominant allele being scored as a recessive allele.
`E2`	if Wang's (2004) or CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers are used this is the probability of an allele being miss-scored. In the CERVUS model errors are not independent for the two alleles within a genotype and so if a genotyping error has occurred at one allele then a genotyping error occurs at the other allele with probability one. Accordingly, `E2`(2-`E2`) is the per-genotype rate defined in CERVUS. If Hadfield's (2009) model of genotyping error for dominant markers is used this is the probability of a recessive allele being scored as a dominant allele.
`ped`	pedigree in 3 columns: id, dam, sire. Base individuals have NA as parents. All parents must be in id.

no_dup	integer: number of times genotypes are to be observed
prop.missing	proportion of observed genotypes that are missing
marker.type	"MSW" or "MSC" for co-dominant markers with Wang's (2004) model of genotyping error or CERVUS's model of genotyping error (Kalinowski, 2006; Marshall, 1998) or "AFLP" for dominant markers (Hadfield, 2009).

Value

G	list of genotype objects; true genotypes for each locus
Gid	vector of id names indexing G
Gobs	list of genotype objects; observed genotypes for each locus
id	vector of id names indexing Gobs

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Marshall, T. C. *et al* (1998) Molecular Ecology 7 5 639-655 Kalinowski S.T. *et al* (2007) Molecular Ecology 16 5 1099-1106 Hadfield J. D. *et al* (2009) *in prep*

See Also

genotype

Examples

```
pedigree<-cbind(1:10, rep(NA,10), rep(NA, 10))

gen_data<-simgenotypes(A=list(loc_1=c(0.5, 0.2, 0.1, 0.075, 0.025)),
 E1=0.1, E2=0.1, ped=pedigree, no_dup=1)

summary(gen_data$G[[1]])
summary(gen_data$Gobs[[1]])
```

---

simpedigree *Simulates a Pedigree given a Log-Linear Model*

Description

Given a PdataPed object simulates a pedigree according to the linear model defined by formula and user specified parameter values for the given model.

Usage

```
simpedigree(PdP, beta=NULL, nUS=NULL)
```

Arguments

PdP	a PdataPed object
beta	parameter vector for the model defined by the formula argument in PdataPed
nUS	vector for the size of the unsampled population(s) defined in the USdam and USsire arguments passed to PdataPed. Parmeters for the unsampled female population come before the male population.

Value

ped	pedigree in 3 columns: id, dam, sire. Base individuals have NA as parents
USsire.data	binary vector indicating unsampled sire records (1)
USsire.formula	variable of the form expression(varPed(...)) that can be included in the formula argument of PdataPed so that unobserved male records are effectively hidden
USdam.data	binary vector indicating unsampled dam records (1)
USdam.formula	variable of the form expression(varPed(...)) that can be included in the formula argument of PdataPed so that unobserved male records are effectively hidden

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31

See Also

MCMCped

---

startPed *startPed Object*

Description

An object containing the starting parameterisation of a model, and logical variables indicating wether parameters should be estimated or fixed at the starting parameterisation. By default the starting parameterisation is obtained through a mixture of Maximum Likelihood and heuristic techniques.

Usage

```
startPed(G=NULL, id=NULL, estG=TRUE, A=NULL, estA=TRUE, E1=NULL,
   estE1=TRUE, E2=NULL, estE2=TRUE, ped=NULL, estP=TRUE,
   beta=NULL, estbeta=TRUE,  USdam=NULL, estUSdam=TRUE,
   USsire=NULL, estUSsire=TRUE, shrink=NULL)
```

Arguments

G	list of genotype objects
id	vector of indivual id's for G
estG	logical; should genotypes be estimated?
A	list of allele frequencies
estA	logical; should base-population allele frequencies be estimated?
E1	if Wang's (2004) model of genotyping error for co-dominant markers is used this is a vector of probabilities of an allele dropping out. If CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers is used this parameter is not used. If Hadfield's (2009) model of genotyping error for dominant markers is used this is a vector of probabilities of a dominant allele being scored as a recessive allele. Default=0.005.
estE1	logical; should E1 estimated?
E2	if Wang's (2004) or CERVUS's (Kalinowski, 2006; Marshall, 1998) model of genotyping error for co-dominant markers are used this is a vector of probabilities of an allele being miss-scored. In the CERVUS model errors are not independent for the two alleles within a genotype and so if a genotyping error has occurred at one allele then a genotyping error occurs at the other allele with probability one. Accordingly, E2(2-E2) is the per-genotype rate defined in CERVUS. If Hadfield's (2009) model of genotyping error for dominant markers is used this is a vector of probabilities of a recessive allele being scored as a dominant allele. Default=0.005.
estE2	logical; should E2 be estimated?
ped	pedigree in 3 columns: id, dam, sire. Base individuals have NA as parents.
estP	logical; should the pedigree be estimated?
beta	vector of population-level parameters
estbeta	logical; should the population-level parameters be estimated?
USdam	vector of unsampled female population sizes
estUSdam	logical; should the female population sizes be estimated?
USsire	vector of unsampled male population sizes
estUSsire	logical or character; if TRUE the male population size is estimated separately from the female population size, if "USdam" male and female population sizes are constrained to be the same.
shrink	optional scalar for the variance defining the ridge-regression likelihood penalisation used to obatain starting values for beta and/or unsampled population sizes.

Details

If estG=FALSE an approximation is used for genotyping error. In this case error rates and allele frequencies are not estimated but fixed at the starting parameterisation. If indivdiuals have been typed more than once, then the approxiamtion only uses the genotype that first appears in the GdP$G object passed to MCMCped. If A is not specified estimates are taken directly from GdP$G using extractA. If E1 and E2 are not specified they are set to 0.005. Note that if the approximation for genotyping error

is used with codominant markers, Wang's (2005) model is not used, and the CEVUS model (Marshall 1998) is adopted. In this case E2 is the per-allele error rate and E2(2-E2) is the per-genotype error rate used by CERVUS. If dam and sire are not specified the most likely set of parents given the genetic data are used (see MLE.ped). The starting value of beta, if not given, is the MLE of beta given the starting pedigree (see MLE.beta). The starting values of USdam and USsire, if not given, are the MLE based on the genotype data (see MLE.popsize).

Value

list containing the arguments passed

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
# In this example we simulate a pedigree and then fix the
# pedigree and estimate the population level paarmeters

data(WarblerP)

var1<-expression(varPed(c("lat", "long"), gender="Male",
  relational="OFFSPRING"))

# paternity is to be modelled as a function of distance
# between offspring and male territories

res1<-expression(varPed("offspring", restrict=0))

# indivdiuals from the offspring generation are excluded as parents

res2<-expression(varPed("terr", gender="Female", relational="OFFSPRING",
  restrict="=="))

# mothers not from the offspring territory are excluded

PdP<-PdataPed(formula=list(var1,res1,res2), data=WarblerP, USsire=FALSE)
simped<-simpedigree(PdP, beta=-0.25)

# simulate a pedigree where paternity drops with distance (beta=-0.25)

sP<-startPed(ped=simped$ped, estP=FALSE)
model1<-MCMCped(PdP=PdP, sP=sP, nitt=3000, thin=2, burnin=1000)
plot(model1$beta)

# The true underlying value is -0.25
```

```
## End(Not run)
```

---

summary.genotypeD *genotypeD Object*

Description

creates and object containing allele and genotype frequency for genotypeD objects

Usage

```
## S3 method for class 'genotypeD'
summary(object, ...)
```

Arguments

object	genotypeD object
...	other arguments to be passed

Value

locus	locus information field (if present)
allele.names	vector of allele names: 0 and 1
allele.freq	estimated allele frequencies with finite sample size correction (Lynch and Milligan 1994)
genotype.freq	frequencies of observed genotypes (phenotypes)

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Lynch M. and Milligan B.G. (1994) Molecular Ecology 3 91-99

See Also

genotype, summary.genotypeD

Examples

```
## Not run:
l1<-rbinom(100,1,0.5)
l1<-genotypeD(l1)
summary(l1)

## End(Not run)
```

tunePed *tunePed Object*

Description

An object containing scaling constants for the tuning parameters used in the Metropolis-Hastings updates. The tuning parameters should be set so that the Metropolis-Hastings acceptance rates lie between 0.2 and 0.5. Initial tuning parameters for beta and the unsampled population size are obtained from the large sample variance-covariances of the Maximum Likelihood estimates.

Usage

```
tunePed(E1 = NULL, E2 = NULL, beta = NULL, USdam = NULL,
   USsire = NULL)
```

Arguments

E1	vector of scaling parameters for E1
E2	vector of scaling parameters for E2
beta	vector which is multiplied by sqrt(10) to get scaling parameters for beta
USdam	vector which is multiplied by 10 to get scaling parameters for the number of unsampled females
USsire	vector which is multiplied by 10 to get scaling parameters for the number of unsampled males

Details

The proposal distribution for all parameters is the multivariate normal, the variances of which are the large sample variance covariances of the Maximum Likelihood estimates multiplied by the scaling constants. For all parameters except beta, the covariance matrix for the proposal distribution has all off-diagonal elements set to zero. These parameters must be positive and so the proposal distribution is reflected at zero. A diagonal covariance matrices ensures that the proposal distribution remains symetric. For beta the covariances are not constrained at zero, and so the matrices are multiplied by the scaling constants in a way that preserves the correlational structure. The tuning parameters for the error rates are the scaling constants multiplied by 3e-5.

Value

list containing the arguments passed

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

See Also

MCMCped

Examples

```
## Not run:
data(WarblerG)
A<-extractA(WarblerG)

ped<-matrix(NA, 100,3)
ped[,1]<-1:100

G<-simgenotypes(A, ped=ped, E1=0.1, E2=0.001, no_dup=2)
GdP<-GdataPed(G=G$Gobs, id=G$id)

model1<-MCMCped(GdP=GdP, nitt=1500, thin=1, burnin=500)

# The proposal distribution is to conservative for E1
# and the update is accepted about 70% of the time

plot(model1$E1)
autocorr(model1$E1)

# Succesive samples from the posterior distribution are
# strongly autocorrelated.  Should of course run the chain
# for longer with a larger thinning interval, but a greater
# tuning parameter helps (now 3e-4, rather than 3e-5):

model2<-MCMCped(GdP=GdP, tP=tunePed(E1=10), nitt=1500,
  thin=1, burnin=500)

plot(model2$E1)
autocorr(model2$E1)

## End(Not run)
```

---

varPed *Transforms Variables for a Multinomial Log-Linear Model*

Description

Creates offspring specific design matrices the columns of which refer to the explanatory variables of the liner model.

Usage

```
varPed(x, gender=NULL, lag=c(0,0), relational=FALSE,
  lag_relational=c(0,0), restrict=NULL, keep=FALSE,
  USvar=NULL, merge=FALSE, NAvar=NULL)
```

Arguments

x	predictor variable; numeric or factor
gender	the gender of the parent to which x applies
lag	numeric vector of length 2. The time interval over which x is evaluated relative to a record of the offspring.
relational	a character string. If "OFFSPRING", the Euclidean distance between x in the parents and x in the offspring is calculated. If "MATE", the Euclidean distance between x in the two parental sexes is calculated. Specifying "OFFSPRINGV" and "MATEV" is similar, although the signed vector is calculated rather than the Euclidean distance. The signed vector is calculated by substracting offspring phenotype from parental phenotype in the case of "OFFSPRINGV", and by substacting the phenotype of the sex NOT specified in gender from the phenotype of the sex specified in gender, in the case of "MATEV". If x is a factor then both the Euclidean distance and the signed vector are 1 if the factor levels for offspring and parent (or the two parental sexes) match, and zero otherwise. If FALSE, x is untransformed.
lag_relational	numeric vector of length 2. If relational is not FALSE then the time interval over which x is evaluated in the relational category relative to the offspring record.
restrict	character string designating parents with a zero prior probability of parentage. Only parents for which x matches restrict have non-zero probabilities of parentage. When relational="OFFSPRING" is specified, then restrict can take on the inequalities "==", "!=", ">", ">=", "<" and "<=". Parents for which the inequalities are satisfied have non-zero probabilities of parentage, with the parental value of x on the left hand side of the inequality and the offspring value on the right hand side. If a number appears on the right hand side of the inequality (e.g. "<=10") then the distance between parent and offspring appears on the left-hand side of the inequality. Restrict is not implemented when relational="MATE"
keep	logical; if TRUE then the design matrices for parents excluded using the argument restrict are retained in the estimation of beta
USvar	if NULL, the phenotypes of unsampled parents are assumed to be drawn from the same statistaical population as the sampled parents. If x is a factor then USvar can be a level of that factor to which unsampled parents belong. If x is numeric then USvar can be the value for unsampled parents. Sampled individuals for which there are missing covariate data will also take on USvar if specified.
merge	logical; if TRUE then beta is the log odds ratio of an offspring's parent belonging to category A compared to category B, where A and B are levels of x. If FALSE then beta is the log odds ratio of an individual belonging to category A being the parent of an offspring compared to an individual of category B. When relational=="MATE", relational=="MATEV" or male and female variables are interacted keep must be FALSE.
NAvar	numeric; replacement for missing values in the predictors.

Details

The design matrix for each offspring represents the state of each parental (dam/sire) combination for each explanatory variable. The number of rows in the design matrix (the number of parental combinations) is free to vary across offspring, but the number of explanatory variables remain the same. As with standard generalised linear modelling the columns of the design matrices take on numerical values or inidicator values for continuous and categorical variables, respectively. When `relational=FALSE`, elements of the design matrices refering to specific parental combinations will not vary across offspring (unless longitudinal data are being used) and the associated vector of parameters will relate the explanatory variables to overall fecundity. For these variables the model is essentially the multinomial analogue of the more familiar Poisson model often used to analyse such data. However, the counts of the multinomial are not known with certainty because uncertainty exists around the maternity and/or paternity of each offspring.

Additional variables can be fitted that relate specific parental combinations to specific offspring, or specific dams to specific sires. Elements of the design matrices refering to specific parental combinations are then free to vary across offspring. The most obvious variable of this type is the mendelian transition probability obtained from the genetic data themsleves. However, by specifying `relational="OFFSPRING"`, `relational="OFFSPRINGV"`, `relational="MATE"` or `relational="MATEV"`, non-genetic variables are free to vary across offspring. When `x` is `numeric` the Euclidean distances between parents and offspring, or between mates enter into the design matrix, when `relational="OFFSPRING"` or `relational="MATE"` respectively. When `relational="OFFSPRINGV"` or `relational="MATEV"` are specified a signed vector is calculated rather than a distance. When `x` is a `factor` then an indicator variable is set up indicating whether parent and offspring, or mate, factor levels match. Often, each offspring will have a variable number of candidate parents as some parents may be excluded *a priori*. When `x` is a `factor` and both `relational="OFFSPRING"` and `restrict="=="`, only those potential parents that have factor levels matching the offspring factor level are retained. When `relational=FALSE`, `restrict` can take on factor levels which exclude parents that have non-matching factor levels.

If a time variable (`timevar`) is not passed to `PdataPed` the data are assumed to be cross-sectional and each indivdiual only respresented once. If a time variable (`timevar`) is passed to `PdataPed` then `lag` and `lag_relational` can be set so that time specific covariates are used. `lag` designates time units relative to the offspring record when `relational=FALSE`; for example, if `lag=c(0,0)` the value of `x` is taken for that parent during the same time period as the offspring record. If `relational="OFFSPRING"` or `relational="MATE"` then `lag` determines the time units relative to the record of the offspring or mate to which the focal inidvidual is being compared. This record can be specified by using `lag_relational`, which is always relative to the offspring record. Negative lags refer to previous time intervals (e.g. `lag=c(-1,-1)` takes `x` from the previous time step), and if the elements of `lag` or `lag_relational` differ then the average value of `x` during this period is taken (e.g `lag=c(-1,0)` averages `x` in the record matching and preceding the offspring record). This is not applicable when `x` is a `factor` unless `restrict` takes one of the logical values (e.g.`"=="`) in which case parents are retained when the logical value is `TRUE` at least once in the specified interval.

Below are models that can be fitted using `varPed`, where `x` is a univariate continuous variable:

`varPed(x, gender="Female")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1 x_i ...)$$

`varPed(x, gender="Male")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1 x_j ...)$$

`varPed(x)`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(x_i + x_j)...)$$

`varPed(x, gender="Female", relational="OFFSPRING")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(|x_i - x_o|)...)$$

`varPed(x, gender="Female", relational="OFFSPRINGV")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(x_i - x_o)...)$$

`varPed(x, gender="Female", relational="MATE")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(|x_i - x_j|)...)$$

`varPed(x, gender="Female", relational="MATEV")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(x_i - x_j)...)$$

`varPed(x, gender="Female", lag=c(-1,-1))`

$$p_{i,j}^{(o)} \propto \exp(\beta_1 x_{i,t-1}...)$$

`varPed(x, gender="Female", lag=c(-1,-1), relational="OFFSPRING")`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(|x_{i,t-1} - x_{o,t}|)...)$$

`varPed(x, gender="Female", lag=c(-2,-2), relational="MATE",`
`lag_relational=c(-1,-1))`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(|x_{i,t-2} - x_{j,t-1}|)...)$$

`varPed(x, gender="Male", lag=c(-2,-2), relational="OFFSPRING",`
`lag_relational=c(-1,-1))`

$$p_{i,j}^{(o)} \propto \exp(\beta_1(|x_{j,t-2} - x_{o,t-1}|)...)$$

Where $p_{i,j}^{(o)}$ is the probability that dam i and sire j are the parents of an offspring o. x and β are the variable of interest and the associated parameter, and t is the time period to which the offspring record belongs.

For a categorical variable with two levels (`A` and `B`) the model specified by `varPed(x, gender="Female")` takes on the form

$$p_{i,j}^{(o)} \propto \exp(\beta_1 \delta_i...)$$

where δ_i is an indicator variable taking the value 1 if x_i is equal to the first level of `x` and zero otherwise. β_1 is then the log odds ratio of the two levels of `x` with respect to maternity. If `merge=TRUE` is specified then β_1 may vary across offspring, and β_m is estimated. β_m is related to β_1:

$$\beta_m = \text{logit}\left[\frac{\theta N_A}{\theta N_A + (1-\theta)N_B}\right]$$

where θ is the inverse logit transformation of β_1, and N_A and N_B are the number of potential mothers that have level `A` and `B` for `x`. If N_A and N_B are invariant over offspring the models are functionally equivalent.

The denominator of the multinomial likelihood is the summed linear predictors of all possible parents (after setting up a contrast with the baseline parents). Designating the first set of parents as baseline, the contrast for each set of parents is simply:

$$\eta_{i,j}^{(o)} = \log\left[\frac{p_{i,j}^{(o)}}{p_{1,1}^{(o)}}\right]$$

and the likelihood of β is

$$Pr(x|\boldsymbol{\beta}) = \prod_{o}^{n_o}\left[\frac{\exp(\eta_{d,s}^{(o)})}{\sum_{i=1}^{n_i^{(o)}}\sum_{j=1}^{n_j^{(o)}}\exp(\eta_{i,j}^{(o)})}\right]$$

where n_o, $n_i^{(o)}$ and $n_j^{(o)}$ are the number of offspring, the number of potential mothers for offspring o, and the number of potential fathers for offspring o, respectively. d and s are the actual parents of offspring o. The set of possible parents in the denominator of the multinomial likelihood are those that are not excluded using the argument `restrict`. However, if the argument `keep=TRUE` is used then the denominator of the likelihood will include excluded parents depsite the fact that $d \neq i$ and $s \neq j$.

In version 2.31-2.42 `DSapprox=TRUE` can be passed to `MCMCped` which approximates the likelihood of β when a variable specifies the distance between mates (i.e `relational="MATE"`). This approximation reduces the computational burden by fixing $i = d$ or $j = s$ in the denominator of the multinomial likelihood. The parent defined as the `"MATE"` is fixed, so that a `varPed` expression with `gender="Male"` has the approximated likelihood:

$$Pr(x|\boldsymbol{\beta}) \approx \prod_{o}^{n_o}\left[\frac{\exp(\eta_{d,s}^{(o)})}{\sum_{j=1}^{n_j^{(o)}}\exp(\eta_{d,j}^{(o)})}\right]$$

For certain types of problem this approximation does not work well. In version 2.43 and after, another approximation is used which seems to work better:

$$Pr(x|\boldsymbol{\beta}) \approx \prod_{o}^{n_o}\left[\frac{\exp(\eta_{d,s}^{(o)})}{\sum_{i=1}^{n_i^{(o)}}\exp(\eta_{i,s}^{(o)}) + \sum_{j=1}^{n_j^{(o)}}\exp(\eta_{d,j}^{(o)}) - \exp(\eta_{d,s}^{(o)})}\right]$$

Value

list containing the design matrix for variable `x`, the identity of retained parents and the gender of the parents

Note

Versions >=2.1 accept different arguments for `restrict` than earlier versions. When `relational="OFFSPRING"`, earlier versions accepted `restrict=TRUE` and `restrict=FALSE`, but these have now been replaced with `restrict="=="` and `restrict="!="`, respectively. In addition, `restrict` now also accepts `">"`, `">="`, `"<"` and `"<="` with parental values on the LHS and offspring values on the RHS.

Also, versions >=2.1 also accept `"OFFSPRINGV"` and `"MATEV"` for `relational` in addition to `"OFFSPRING"` and `"MATE"`. `"V"` specifies that the signed vector should be used rather than the Euclidean distance.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk>

References

Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31

See Also

MCMCped

`WarblerG` *Seychelles Warbler Genotypes*

Description

Genetype data collected by David Richardson from Cousin Island in 1999.

Format

a data frame with 307 rows and 29 columns. The first column are the unique idenitifiers for each bird, and the following columns are genotype data. Adjacent columns beolng to the same locus.

Source

Richardson D.S.

References

Richardson *et.al.* (2001) Molecular Ecology 10 2263-2273 Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31

`WarblerP` *Seychelles Warbler Phenotypes*

Description

Phenotypic data collected by David Richardson from Cousin Island in 1999. The data are almost a complete sample of those birds that existed in the population at that time.

Format

a table with 307 rows and 7 columns. The columns, from left to right are: 1) a unique identifier for each bird; 2) a binary variable inbdicating whether the record belongs to an offspring; 3) the sex of each bird; 4) the territory on which the bird was recorded; 5 and 6) the latitude and longitude of that territory; 7) the behavioural status of each bird (Dominant or Subordinate)

Source

Richardson D.S.

References

Richardson *et.al.* (2001) Molecular Ecology 10 2263-2273 Hadfield J.D. *et al* (2006) Molecular Ecology 15 3715-31

Index