

Package ‘MCMCglmm’

November 12, 2009

Title MCMC Generalised Linear Mixed Models

Version 1.13

Depends MASS, corpcor, tensorA, Matrix, coda, ape, pscl, orthopolynom, combinat, gtools

Suggests rgl

Date 2009-11-12

Author Jarrod Hadfield

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

Description MCMC Generalised Linear Mixed Models

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-11-12 15:58:06

R topics documented:

MCMCglmm-package	2
at.level	3
at.set	4
BTdata	4
BTped	5
commutation	6
Ddivergence	6
Dexpressions	7
Dtensor	8
evalDtensor	9
inverseA	10
knorm	11
KPPM	12
krzanowski.test	12
kunif	13

leg	14
list2bdiag	15
MCMCglimm	16
PlodiaPO	19
PlodiaR	19
PlodiaRB	20
plotprior	21
plotsubspace	22
posterior.cor	23
posterior.evals	23
posterior.mode	24
prunePed	25
Ptensor	26
rbv	26
rIW	27
sm2asreml	29
SShorns	30
Tri2M	30
Index	32

MCMCglimm-package *Multivariate Generalised Linear Mixed Models*

Description

MCMCglimm is a package for fitting Generalised Linear Mixed Models using Markov chain Monte Carlo techniques (Hadfield 2009). Most commonly used distributions like the normal and the Poisson are supported together with some useful but less popular ones like the zero-inflated Poisson and the multinomial. Missing values and left, right and interval censoring are accommodated for all traits. The package also supports multi-trait models where the multiple responses can follow different types of distribution. The package allows various residual and random-effect variance structures to be specified including heterogeneous variances, unstructured covariance matrices and random regression (e.g. random slope models). Three special types of variance structure that can be specified are those associated with pedigrees (animal models), phylogenies (the comparative method) and measurement error (meta-analysis).

The package makes heavy use of results in Sorensen & Gianola (2002) and Davis (2006) which taken together result in what is hopefully a fast and efficient routine. Most small to medium sized problems should take seconds to a few minutes, but large problems (> 20,000 records) are possible. My interest is in evolutionary biology so there are also several functions for applying Rice's (2004) tensor analysis to real data and functions for visualising and comparing matrices.

Please read the tutorial: `vignette("Tutorial", "MCMCglimm")`

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

- Hadfield, J.D. (2009) MCMC methods for Multi-response Generalised Linear Mixed Models: The MCMCglmm R Package, *submitted*
- Sorensen, D. & Gianola, D. (2002) Likelihood, Bayesian and MCMC Methods in Quantitative Genetics, Springer
- Davis, T.A. (2006) Direct Methods for Sparse Linear Systems, SIAM
- Rice (2004) Evolutionary Theory: Mathematical and Conceptual Foundations, Sinauer

`at.level`*Incidence Matrix of Levels within a Factor*

Description

Incidence matrix of levels within a factor

Usage

```
at.level(x, level)
```

Arguments

<code>x</code>	factor
<code>level</code>	factor level

Value

incidence matrix for level in x

Author(s)

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

See Also

[at.set](#)

Examples

```
fac<-gl(3,10,30, labels=letters[1:3])
x<-rnorm(30)
model.matrix(~at.level(fac, "b"):x)
```

`at.set`*Incidence Matrix of Combined Levels within a Factor*

Description

Incidence Matrix of Combined Levels within a Factor

Usage

```
at.set(x, level)
```

Arguments

<code>x</code>	factor
<code>level</code>	set of factor levels

Value

incidence matrix for the set `level` in `x`

Author(s)

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

See Also

[at.level](#)

Examples

```
fac<-gl(3,10,30, labels=letters[1:3])
x<-rnorm(30)
model.matrix(~at.set(fac,2:3):x)
```

`BTdata`*Blue Tit Data for a Quantitative Genetic Experiment*

Description

Blue Tit (*Cyanistes caeruleus*) Data for a Quantitative Genetic Experiment

Usage

```
BTdata
```

Format

a data frame with 828 rows and 7 columns, with variables tarsus length (`tarsus`) and colour (`back`) measured on 828 individuals (`animal`). The mother of each is also recorded (`dam`) together with the foster nest (`fosternest`) in which the chicks were reared. The date on which the first egg in each nest hatched (`hatchdate`) is recorded together with the sex (`sex`) of the individuals.

References

Hadfield, J.D. et. al. 2007 *Journal of Evolutionary Biology* 20 549-557

See Also

[BTped](#)

BTped

Blue Tit Pedigree

Description

Blue Tit (*Cyanistes caeruleus*) Pedigree

Usage

BTped

Format

a data frame with 1040 rows and 3 columns, with individual identifier (`animal`) mother identifier (`dam`) and father identifier (`sire`). The first 212 rows are the parents of the 828 offspring from 106 full-sibling families. Parents are assumed to be unrelated to each other and have NA's in the `dam` and `sire` column.

References

Hadfield, J.D. et. al. 2007 *Journal of Evolutionary Biology* 20 549-557

See Also

[BTped](#)

commutation	<i>Commutation Matrix</i>
-------------	---------------------------

Description

Forms an $m \times m$ commutation matrix which transforms $\text{vec}(\mathbf{A})$ into $\text{vec}(\mathbf{A}')$, where \mathbf{A} is an $m \times n$ matrix

Usage

```
commutation(m, n)
```

Arguments

m	integer; number of rows of A
n	integer; number of columns of A

Value

Commutation Matrix

Author(s)

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

References

Magnus, J. R. & Neudecker, H. (1979) *Annals of Statistics* 7 (2) 381-394

Examples

```
commutation(2,2)
```

Ddivergence	<i>d-divergence</i>
-------------	---------------------

Description

Calculates Ovaskainen's (2008) d-divergence between 2 zero-mean multivariate normal distributions.

Usage

```
Ddivergence(CA=NULL, CB=NULL, n=10000)
```

Arguments

CA	Matrix A
CB	Matrix B
n	number of Monte Carlo samples for approximating the integral

Value

d-divergence

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

References

Ovaskainen, O. et. al. (2008) Proc. Roy. Soc - B (275) 1635 593-750

Examples

```
CA<-rIW(diag(2),10, n=1)
CB<-rIW(diag(2),10, n=1)
Ddivergence(CA, CB)
```

Dexpressions	<i>List of unevaluated expressions for (mixed) partial derivatives of fitness with respect to linear predictors.</i>
--------------	--

Description

Unevaluated expressions for (mixed) partial derivatives of fitness with respect to linear predictors for survival and fecundity.

Usage

```
Dexpressions
```

Value

PW.d0W	Fitness (W) function for the Poisson-Weibull (PW) model.
PW.d1Wds	First Partial derivative of fitness (d1W) with respect to survival (d1s) linear predictor for the Poisson-Weibull (PW) model.
PW.d1Wdf	First Partial derivative of fitness (d1W) with respect to fecundity (d1f) linear predictor for the Poisson-Weibull (PW) model.
PW.d3Wd2sd1f	Mixed third partial derivative of fitness (d3W) with 2nd derivative of survival linear predictor (d2s) and first derivative of fecundity linear predictor (d1f) from the Poisson-Weibull (PW) model.

PW.d3Wdsd2f and so on ...
 PW.d2Wd2f
 PW.d2Wd2s
 PW.d3Wd3s
 PW.d3Wd3f

Author(s)

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

See Also

[Dtensor](#)

Dtensor *Tensor of (mixed) partial derivatives*

Description

Forms tensor of (mixed) partial derivatives

Usage

```
Dtensor(expr, name=NULL, mu = NULL, m=1, evaluate = TRUE)
```

Arguments

expr	'expression'
name	character vector, giving the variable names with respect to which derivatives will be computed. If NULL all variables in the expression will be used
mu	optional: numeric vector, at which the derivatives are evaluated
m	order of derivative
evaluate	logical; if TRUE the derivatives are evaluated at mu, if FALSE the derivatives are left unevaluated

Value

Dtensor (list) of unevaluated expression(s) if evaluate=FALSE or a tensor if evaluate=TRUE

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

References

Rice, S.H. (2004) Evolutionary Theory: Mathematical and Conceptual Foundations. Sinauer (MA) USA.

See Also

[evalDtensor](#), [Dexpressions](#), [D](#)

Examples

```
f<-expression(beta_1 + time * beta_2 + u)
Dtensor(f, eval=FALSE)
```

evalDtensor

Evaluates a list of (mixed) partial derivatives

Description

Evaluates a list of (mixed) partial derivatives

Usage

```
evalDtensor(x, mu)
```

Arguments

x	unevaluated (list) of expression(s)
mu	values at which the derivatives are evaluated: names need to match terms in x

Value

tensor

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

See Also

[Dtensor](#), [D](#)

Examples

```
f<-expression(beta_1 + time*beta_2+u)
Df<-Dtensor(f, eval=FALSE, m=2)
evalDtensor(Df, mu=data.frame(beta_1=0.5, beta_2=1, time=3, u=2.3))
Dtensor(f, mu=c(1, 3, 1, 2.3), m=2)
```

inverseA

*Inverse Relatedness Matrix and Phylogenetic Covariance Matrix***Description**

Henderson (1976) and Meuwissen and Luo (1992) algorithm for inverting relatedness matrices, and Hadfield and Nakagawa (2009) algorithm for inverting phylogenetic covariance matrices.

Usage

```
inverseA(pedigree=NULL, nodes="ALL", scale=TRUE)
```

Arguments

pedigree	ordered pedigree with 3 columns: id, dam and sire, or a <code>phylo</code> object.
nodes	"ALL" calculates the inverse for all individuals/nodes. For phylogenies "TIPS" calculates the inverse for the species tips only, and for pedigrees a vector of id's can be passed which inverts the relatedness matrix for that subset.
scale	logical: should a phylogeny (needs to be ultrametric) be scaled to unit length (distance from root to tip)?

Value

Ainv	inverse as <code>sparseMatrix</code>
inbreeding	inbreeding coefficients/branch lengths
pedigree	pedigree/pedigree representation of phylogeny

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

Henderson, C.R. (1976) *Biometrics* 32 (1) 69:83
 Meuwissen, T.H.E, and Luo, Z. (1992) 24 (4) 305:313
 Hadfield, J.D. and Nakagawa, S. (2009) *submitted*

Examples

```
data(bird.families)
Ainv<-inverseA(bird.families)
```

`knorm`*(Mixed) Central Moments of a Multivariate Normal Distribution*

Description

Forms a tensor of (mixed) central moments of a multivariate normal distribution

Usage

```
knorm(V, k)
```

Arguments

V	(co)variance matrix
k	kth central moment, must be even

Value

tensor

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

References

Schott, J.R.(2003) *Journal of Multivariate Analysis* 87 (1) 177-190

See Also

[dnorm](#)

Examples

```
V<-diag(2)
knorm(V, 2)
knorm(V, 4)
```

KPPM

Kronecker Product Permutation Matrix

Description

Forms an $m \times k$ Kronecker Product Permutation Matrix

Usage

```
KPPM(m, k)
```

Arguments

m integer

k integer

Value

Matrix

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

Schott, J.R.(2003) Journal of Multivariate Analysis 87 (1) 177-190

Examples

```
KPPM(2, 3)
```

krzanowski.test*Krzanowski's Comparison of Subspaces*

Description

Calculates statistics of Krzanowski's comparison of subspaces.

Usage

```
krzanowski.test(CA, CB, vecsA, vecsB, corr = FALSE, ...)
```

Arguments

CA	Matrix A
CB	Matrix B
vecsA	Vector of integers indexing the eigenvectors determining the subspace of A
vecsB	Vector of integers indexing the eigenvectors determining the subspace of B
corr	logical; if TRUE the variances of A and B are standardised
...	further arguments to be passed

Value

sumofS	metric for overall similarity with 0 indicating no similarity and a value of length (vecsA) for identical subspaces
angles	angle in degrees between each best matched pair of vectors
bisector	vector that lies between each best matched pair of vectors

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

Krzanowski, W.J. (2000) Principles of Multivariate Analysis. OUP

Examples

```
CA<-rIW(diag(5),10, n=1)
CB<-rIW(diag(5),10, n=1)
krzanowski.test(CA, CB, vecsA=1:2, vecsB=1:2)
krzanowski.test(CA, CA, vecsA=1:2, vecsB=1:2)
```

kunif

Central Moments of a Uniform Distribution

Description

Returns the central moments of a uniform distribution

Usage

```
kunif(min, max, k)
```

Arguments

min, max	lower and upper limits of the distribution. Must be finite.
k	k central moment, must be even

Value

kth central moment

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

See Also

[dunif](#)

Examples

```
kunif(-1, 1, 4)
y<-runif(1000, -1, 1)
mean((y-mean(y))^4)
```

leg

Evaluates Orthogonal Legendre Polynomials

Description

Evaluates orthogonal Legendre polynomials of degree 1 to degree over the specified set of points `x`

Usage

```
leg(x, degree, normalized=TRUE)
```

Arguments

<code>x</code>	a numeric vector at which to evaluate the polynomial
<code>degree</code>	the degree of the polynomial. If negative, degree 0 is not evaluated
<code>normalized</code>	logical: should the polynomial be normalised

Details

Please refer to [legendre.polynomials](#) in the package `orthopolynom` for more details.

Value

Matrix of evaluated polynomials

Author(s)

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

See Also

[legendre.polynomials](#)

Examples

```
x<-rnorm(30)
model.matrix(~leg(x,2)-1)
```

list2bdiag

Forms the direct sum from a list of matrices

Description

Forms a block-diagonal matrix from a list of matrices

Usage

```
list2bdiag(x)
```

Arguments

x list of square matrices

Value

matrix

Author(s)

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

Examples

```
M<-list(rIW(diag(3), 10), rIW(diag(2), 10))
list2bdiag(M)
```

Description

Markov chain Monte Carlo Sampler for Multivariate Generalised Linear Mixed Models with special emphasis on correlated random effects arising from pedigrees and phylogenies (Hadfield 2009). Please read the tutorial: `vignette("Tutorial", "MCMCglmm")`

Usage

```
MCMCglmm(fixed, random=NULL, rcov=~units, family="gaussian", mev=NULL,
  data,start=NULL, prior=NULL, tune=NULL, pedigree=NULL,
  nodes="ALL", scale=TRUE, nitt=13000, thin=10, burnin=3000, pr=FALSE,
  pl=FALSE, verbose=TRUE, DIC=TRUE, singular.ok=FALSE, saveX=FALSE, saveZ=FALSE)
```

Arguments

<code>fixed</code>	<code>formula</code> for the fixed effects, multiple responses are passed as a matrix using <code>cbind</code>
<code>random</code>	<code>formula</code> for the random effects. Multiple random terms can be passed using the <code>+</code> operator, and in the most general case each random term has the form <code>variance.function(formula):random.term</code> . Currently, the only variance.functions available are <code>idv</code> , <code>idh</code> and <code>us</code> . <code>idv</code> fits a constant variance across all components in <code>formula</code> . Both <code>idh</code> and <code>us</code> fit different variances across each component in <code>formula</code> , but <code>us</code> will also fit the covariances. The <code>formula</code> can contain both factors and numeric terms (i.e. random regression) although it should be noted that the intercept term is suppressed. The (co)variances are the (co)variances of the <code>random.term</code> effects. For simple random effects the <code>variance.function(formula)</code> can be omitted and the model syntax has the simpler form <code>~random1+random2+...</code> . There are three reserved variables: <code>units</code> which index rows of the response variable, <code>trait</code> which index columns of the response variable and <code>animal</code> which is associated with a pedigree or phylogeny.
<code>rcov</code>	<code>formula</code> for residual covariance structure. This has to be set up so that each data point is associated with a unique residual. For example a multi-response model might have the R-structure defined by <code>~us(trait):units</code>
<code>family</code>	optional character vector of trait distributions. Currently, "gaussian", "poisson", "categorical", "multinomial", "ordinal", "ordinal", "exponential", "cengaussian", "cenpoisson", "cenexponential", "zipoisson" are supported, where the prefix "cen" means censored and the prefix "zi" means zero inflated. If NULL data needs to contain a family column.
<code>mev</code>	optional vector of measurement error variances for each data point for random effect meta-analysis.
<code>data</code>	<code>data.frame</code>

start	optional list having 4 possible elements: R (R-structure) G (G-structure) and Liab (latent variables or liabilities) should contain the starting values where G itself is also a list with as many elements as random effect components. The fourth element QUASI should be logical: if TRUE starting latent variables are obtained heuristically, if FALSE then they are sampled from a Z-distribution
prior	optional list of prior specifications having 3 possible elements: R (R-structure) G (G-structure) and B (fixed effects). Each element is a list containing the expected (co)variances (V) and a degree of belief parameter (nu) for the variance structures (R and G). B is a list containing the expected value (mu) and a (co)variance matrix (V) representing the strength of belief. Defaults to nu=0 and B\$V<-I*1e+8, where I is an identity matrix of appropriate dimension.
tune	optional (co)variance matrix defining the proposal distribution for the latent variables. If NULL an adaptive algorithm is used which ceases to adapt once the burn-in phase has finished.
pedigree	ordered pedigree with 3 columns id, dam and sire or a phylo object.
nodes	pedigree/phylogeny nodes to be estimated. The default, "ALL" estimates effects for all individuals in a pedigree or nodes in a phylogeny (including ancestral nodes). For phylogenies "TIPS" estimates effects for the tips only, and for pedigrees a vector of ids can be passed to nodes specifying the subset of individuals for which animal effects are estimated. Note that all analyses are equivalent if omitted nodes have missing data but by absorbing these nodes the chain mix better. However, the algorithm may be less numerically stable and may iterate slower, especially for large phylogenies.
scale	logical: should the phylogeny (needs to be ultrametric) be scaled to unit length (distance from root to tip)?
nitt	number of MCMC iterations
thin	thinning interval
burnin	burnin
pr	logical: should the posterior distribution of random effects be saved?
pl	logical: should the posterior distribution of latent variables be saved?
verbose	logical: if TRUE MH diagnostics are printed to screen
DIC	logical: if TRUE deviance and deviance information criterion are calculated
singular.ok	logical: if FALSE linear dependencies in the fixed effects are removed. if TRUE they are left in an estimated, although all information comes from the prior
saveX	logical: save fixed effect design matrix
saveZ	logical: save random effect design matrix

Value

Sol	Posterior Distribution of MME solutions, including fixed effects
VCV	Posterior Distribution of (co)variance matrices
Liab	Posterior Distribution of latent variables
Fixed	formula: fixed terms

Random	formula: random terms
Residual	formula: residual terms
Deviance	deviance $-2 \cdot \log(p(yl\dots))$
DIC	deviance information criterion
X	sparse fixed effect design matrix
Z	sparse random effect design matrix

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

General analyses: Hadfield, J.D. (2009) *submitted*
 Phylogenetic analyses: Hadfield, J.D. and Nakagawa, S. (2009) *submitted*
 Background Sorensen, D. & Gianola, D. (2002) Springer

See Also

[mcmc](#)

Examples

```
# Example 1: univariate Gaussian model with standard random effect

data(PlodiaPO)
modell<-MCMCglmm(PO~1, random=~FSfamily, data=PlodiaPO, verbose=FALSE)
plot(modell$VCV)

# Example 2: univariate Gaussian model with phylogenetically correlated
# random effect

data(bird.families)

phylo.effect<-rbv(bird.families, 1, nodes="TIPS")
phenotype<-phylo.effect+rnorm(dim(phylo.effect)[1], 0, 1)

# simulate phylogenetic and residual effects with unit variance

test.data<-data.frame(phenotype=phenotype, animal=row.names(phenotype))

prior<-list(R=list(V=1, nu=1), G=list(G1=list(V=1, nu=1)))

model2<-MCMCglmm(phenotype~1, random=~animal, data=test.data,
pedigree=bird.families, prior=prior, verbose=FALSE)

plot(model2$VCV)
```

 PlodiaPO

Phenoloxidase measures on caterpillars of the Indian meal moth.

Description

Phenoloxidase measures on caterpillars of the Indian meal moth (*Plodia interpunctella*).

Usage

PlodiaPO

Format

a data frame with 511 rows and 3 columns, with variables indicating full-sib family (`FSfamily`), phenoloxidase measures (`PO`), and plate (`plate`). `PO` has undergone a Box-Cox power transformation of 0.141

Source

Tidbury H & Boots M (2007) University of Sheffield

See Also

[PlodiaR](#), [PlodiaRB](#)

 PlodiaR

Resistance of Indian meal moth caterpillars to the granulosis virus PiGV.

Description

Resistance of Indian meal moth (*Plodia interpunctella*) caterpillars to the granulosis virus PiGV.

Usage

PlodiaR

Format

a data frame with 50 rows and 5 columns, with variables indicating full-sib family (`FSfamily`), date of egg laying (`date_laid`) and assaying (`date_Ass`), and the number of individuals from the family that were experimentally infected with the virus `Infected` and the number of those that pupated `Pupated`. These full-sib family identifiers also relate to the full-sib family identifiers in `PlodiaPO`

Source

Tidbury H & Boots M (2007) University of Sheffield

See Also

[PlodiaRB](#), [PlodiaPO](#)

PlodiaRB

Resistance (as a binary trait) of Indian meal moth caterpillars to the granulosis virus PiGV.

Description

Resistance (as a binary trait) of Indian meal moth (*Plodia interpunctella*) caterpillars to the granulosis virus PiGV.

Usage

```
PlodiaRB
```

Format

a data frame with 784 rows and 4 columns, with variables indicating full-sib family (`FSfamily`), date of egg laying (`date_laid`) and assaying (`date_Ass`), and a binary variable indicating whether an individual was resistant (`Pupated`) to an experimental infection of the virus. These data are identical to those in the data.frame `PlodiaR` except each family-level binomial variable has been expanded into a binary variable for each individual.

Source

Tidbury H & Boots M (2007) University of Sheffield

See Also

[PlodiaR](#), [PlodiaPO](#)

plotprior	<i>Histogram of draws from the prior distribution for the variance structure of an MCMCglmm model</i>
-----------	---

Description

Samples from the prior distribution of the R-structure and G-structure used in MCMCglmm models, and plots the histograms

Usage

```
plotprior(prior, cor=FALSE, n=1000, ask = dev.interactive(), ...)
```

Arguments

prior	list of prior variance-structure specifications having 2 possible elements: the R (R-structure) and G (G-structure) specifying the expected (co)variances (V) and degree of belief parameter
cor	logical; if TRUE, the prior correlation matrices are generated
n	integer: number of samples to be drawn
ask	logical: prompt user before each page of plots
...	Further arguments

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[rIW](#)

Examples

```
prior=list(R=list(V=diag(2), n=2),
           G=list(G1=list(V=diag(2), n=3),
                 G2=list(V=diag(2), n=4)))
plotprior(prior, cor=TRUE)
plotprior(prior, cor=FALSE, xlim=c(-1,5))
```

plotsubspace *Plots covariance matrices*

Description

Represents covariance matrices as 3-d ellipsoids using the `rgl` package. Covariance matrices of dimension greater than 3 are plotted on the subspace defined by the first three eigenvectors.

Usage

```
plotsubspace(CA, CB=NULL, corr = FALSE, shadeCA = TRUE,
             shadeCB = TRUE, axes.lab = FALSE, ...)
```

Arguments

CA	Matrix
CB	Optional second matrix
corr	If TRUE the covariance matrices are transformed into correlation matrices
shadeCA	If TRUE the ellipsoid is solid, if FALSE the ellipsoid is wireframe
shadeCB	If TRUE the ellipsoid is solid, if FALSE the ellipsoid is wireframe
axes.lab	If TRUE the axes are labelled with the eigenvectors
...	further arguments to be passed

Details

The matrix CA is always red, and the matrix CB if given is always blue. The subspace is defined by the first three eigenvectors of CA, and the percentage of variance for each matrix along these three dimensions is given in the plot title.

Author(s)

Jarrold Hadfield <j.hadfield@ed.ac.uk> with code taken from the `rgl` package

See Also

[rgl](#)

Examples

```
if(require(rgl)!=FALSE){
  G1<-rIW(diag(4),10)
  G2<-G1*1.2
  plotsubspace(G1, G2, shadeCB=FALSE)
}
```

posterior.cor *Transforms posterior distribution of covariances into correlations*

Description

Transforms posterior distribution of covariances into correlations

Usage

```
posterior.cor(x)
```

Arguments

x mcmc object of (co)variances stacked column-wise

Value

posterior correlation matrices

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[posterior.evals](#)

Examples

```
v<-rIW(diag(2),3, n=1000)
hist(posterior.cor(mcmc(v))[,2])
```

posterior.evals *Posterior distribution of eigenvalues*

Description

Posterior distribution of eigenvalues

Usage

```
posterior.evals(x)
```

Arguments

x mcmc object of (co)variances stacked column-wise

Value

posterior eigenvalues

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[posterior.cor](#)

Examples

```
v<-rIW(diag(2),3, n=1000)
hist (posterior.evals (mcmc (v)) [,2])
```

posterior.mode	<i>Estimates the marginal parameter modes using kernel density estimation</i>
----------------	---

Description

Estimates the marginal parameter modes using kernel density estimation

Usage

```
posterior.mode(x, adjust=0.1, ...)
```

Arguments

x	mcmc object
adjust	numeric, passed to density to adjust the bandwidth of the kernel density
...	other arguments to be passed

Value

modes of the kernel density estimates

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[density](#)

Examples

```
v<-rIW(as.matrix(1),10, n=1000)
hist(v)
abline(v=posterior.mode(mcmc(v)), col="red")
```

prunePed

Pedigree pruning

Description

Creates a subset of a pedigree by retaining the ancestors of a specified subset of individuals

Usage

```
prunePed(pedigree, keep, make.base=FALSE)
```

Arguments

pedigree	pedigree with id in column 1 dam in column 2 and sire in column 3
keep	individuals in pedigree for which the ancestors should be retained
make.base	logical: should ancestors that do not provide additional information be discarded?

Value

subsetting pedigree

Note

If the individuals in `keep` are the only phenotyped individuals for some analysis then some non-phenotyped individuals can often be discarded if they are not responsible for pedigree links between phenotyped individuals. In the simplest case (`make.base=FALSE`) all ancestors of phenotyped individuals will be retained, although further pruning may be possible using `make.base=TRUE`. In this case all pedigree links that do not connect phenotyped individuals are discarded resulting in some individuals becoming part of the base population. In terms of variance component and fixed effect estimation pruning the pedigree should have no impact on the target posterior distribution, although convergence and mixing may be better because there is less missing data.

Author(s)

Jarrod Hadfield <j.hadfield@ed.ac.uk> + Michael Morrissey

Ptensor

Tensor of Sample (Mixed) Central Moments

Description

Forms a tensor of sample (mixed) central moments

Usage

```
Ptensor(x, k)
```

Arguments

x	matrix; traits in columns samples in rows
k	kth central moment

Value

tensor

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

Examples

```
n<-1000
y<-mvrnorm(n, c(0,0), diag(2))
Ptensor(y,2)
cov(y)*((n-1)/n)
```

rbv

Random Generation of MVN Breeding Values and Phylogenetic Effects

Description

Random Generation of MVN Breeding Values and Phylogenetic Effects

Usage

```
rbv(pedigree, G, nodes="ALL", scale=TRUE, ggroups=NULL, gmeans=NULL)
```

Arguments

pedigree	ordered pedigree with 3 columns id, dam and sire or a <code>phylo</code> object.
G	(co)variance matrix
nodes	effects for pedigree/phylogeny nodes to be returned. The default, <code>nodes="ALL"</code> returns effects for all individuals in a pedigree or nodes in a phylogeny (including ancestral nodes). For phylogenies <code>nodes="TIPS"</code> returns effects for the tips only, and for pedigrees a vector of ids can be passed to <code>nodes</code> specifying the subset of individuals for which animal effects are returned.
scale	logical: should a phylogeny (needs to be ultrametric) be scaled to unit length (distance from root to tip)?
ggroups	optional; vector of genetic groups
gmeans	matrix of mean breeding value for genetic groups (rows) by traits (columns)

Value

matrix of breeding values/phylogenetic effects

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

Examples

```
data(bird.families)
bv<-rbv(bird.families, diag(2))
```

rIW	<i>Random Generation from the Conditional Inverse Wishart Distribution</i>
-----	--

Description

Samples from the inverse Wishart distribution, with the possibility of conditioning on a diagonal submatrix

Usage

```
rIW(V, nu, fix=NULL, n=1, CM=NULL)
```

Arguments

V	Expected (co)varaince matrix as <code>nu</code> tends to infinity
nu	degrees of freedom
fix	optional integer indexing the partition to be conditioned on
n	integer: number of samples to be drawn
CM	matrix: optional matrix to condition on. If not given, and <code>fix!=NULL</code> , <code>V_22</code> is conditioned on

Details

If \mathbf{W}^{-1} is a draw from the inverse Wishart, `fix` indexes the diagonal element of \mathbf{W}^{-1} which partitions \mathbf{W}^{-1} into 4 submatrices. `fix` indexes the upper left corner of the lower diagonal matrix and it is this matrix that is conditioned on.

For example partitioning \mathbf{W}^{-1} such that

$$\mathbf{W}^{-1} = \begin{bmatrix} \mathbf{W}^{-1}_{11} & \mathbf{W}^{-1}_{12} \\ \mathbf{W}^{-1}_{21} & \mathbf{W}^{-1}_{22} \end{bmatrix}$$

`fix` indexes the upper left corner of \mathbf{W}^{-1}_{22} . If `CM!=NULL` then \mathbf{W}^{-1}_{22} is fixed at `CM`, otherwise \mathbf{W}^{-1}_{22} is fixed at V_{22} . For example, if `dim(V)=4` and `fix=2` then \mathbf{W}^{-1}_{11} is a 1X1 matrix and \mathbf{W}^{-1}_{22} is a 3X3 matrix.

Value

if `n = 1` a matrix equal in dimension to `V`, if `n>1` a matrix of dimension `n x length(V)`

Note

In versions of `MCMCglmm >1.10` the arguments to `rIW` have changed so that they are more intuitive in the context of `MCMCglmm`. Following the notation of Wikipedia (http://en.wikipedia.org/wiki/Inverse-Wishart_distribution) the inverse scale matrix $\Psi = (V * nu)$. In earlier versions of `MCMCglmm (<1.11)` $\Psi = V^{-1}$. Although the old parameterisation is consistent with the `riwish` function in `MCMCpack` and the `rwishart` function in `bayesm` it is inconsistent with the prior definition for `MCMCglmm`. The following pieces of code are sampling from the same distributions:

<code>riwish(nu, nu*V)</code>	from <code>MCMCpack</code>
<code>rwishart(nu, solve(nu*V))\$IW</code>	from <code>bayesm</code>
<code>rIW(nu, solve(nu*V))</code>	from <code>MCMCglmm <1.11</code>
<code>rIW(V, nu)</code>	from <code>MCMCglmm >=1.11</code>

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

Korsgaard, I.R. et. al. 1999 Genetics Selection Evolution 31 (2) 177:181

See Also

`rwishart`, `rwish`

Examples

```
nu<-10
```

```
V<-diag(4)
rIW(V, nu, fix=2)
```

`sm2asreml`*Converts sparseMatrix to asreml's giv format*

Description

Converts sparseMatrix to asreml's giv format: row-ordered, upper triangle sparse matrix.

Usage

```
sm2asreml(A, rownames)
```

Arguments

<code>A</code>	sparseMatrix
<code>rownames</code>	rownames of A

Value

data.frame: if A was formed from a pedigree equivalent to giv format returned by asreml . Ainverse

Author(s)

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

See Also

inverseA

Examples

```
data(bird.families)
A<-inverseA(bird.families)
Aasreml<-sm2asreml(A$Ainv, A$node.names)
```

 SShorns

Horn type and genders of Soay Sheep

Description

Horn type and genders of Soay Sheep *Ovis aires*

Usage

BTdata

Format

a data frame with 666 rows and 3 columns, with individual identifier (`id`), horn type (`horn`) and gender (`sex`).

References

Clutton-Brock T., Pemberton, J. Eds. 2004 Soay Sheep: Dynamics & Selection in an Island Population

 Tri2M

Lower/Upper Triangle Elements of a Matrix

Description

Lower/Upper triangle elements of a matrix or forms a matrix from a vector of lower/upper triangle elements

Usage

Tri2M(x, lower.tri = TRUE, reverse = TRUE)

Arguments

<code>x</code>	Matrix or vector
<code>lower.tri</code>	If <code>x</code> is a matrix then the lower triangle (TRUE) or upper triangle FALSE elements (including diagonal elements) are returned. If <code>x</code> is a vector a matrix is formed under the assumption that <code>x</code> are the lower triangle (TRUE) or upper triangle (FALSE) elements.
<code>reverse</code>	logical: if TRUE a symmetric matrix is formed, if FALSE the remaining triangle is left as zeros.

Value

numeric or matrix

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

Examples

```
M<-rIW(diag(3), 10)
x<-Tri2M(M)
x
Tri2M(x, reverse=TRUE)
Tri2M(x, reverse=FALSE)
```

Index

- *Topic **array**
 - commutation, 5
 - Dtensor, 8
 - evalDtensor, 9
 - inverseA, 9
 - KPPM, 11
 - Ptensor, 25
- *Topic **datasets**
 - BTdata, 4
 - BTped, 4
 - PlodiaPO, 18
 - PlodiaR, 18
 - PlodiaRB, 19
 - SShorns, 29
- *Topic **distribution**
 - knorm, 10
 - kunif, 13
 - posterior.cor, 22
 - posterior.evals, 22
 - posterior.mode, 23
 - rbv, 25
 - rIW, 26
- *Topic **hplot**
 - plotprior, 20
 - plotsubspace, 21
- *Topic **manip**
 - at.level, 2
 - at.set, 3
 - leg, 13
 - list2bdiag, 14
 - prunePed, 24
 - sm2asreml, 28
 - Tri2M, 29
- *Topic **math**
 - Dtensor, 8
 - evalDtensor, 9
- *Topic **models**
 - MCMCglmm, 15
- *Topic **multivariate**
 - Ddivergence, 6
 - krzanowski.test, 12
- *Topic **symbolmath**
 - Dexpressions, 7
- at.level, 2, 3
- at.set, 3, 3
- BTdata, 4
- BTped, 4, 4, 5
- commutation, 5
- D, 8, 9
- Ddivergence, 6
- density, 23
- Dexpressions, 7, 8
- dnorm, 11
- Dtensor, 7, 8, 9
- dunif, 13
- evalDtensor, 8, 9
- formula, 15
- inverseA, 9
- knorm, 10
- KPPM, 11
- krzanowski.test, 12
- kunif, 13
- leg, 13
- legendre.polynomials, 14
- list2bdiag, 14
- mcmc, 17
- MCMCglmm, 15, 27
- MCMCglmm-package, 1
- PlodiaPO, 18, 19
- PlodiaR, 18, 18, 19

PlodiaRB, [18](#), [19](#), [19](#)
plotprior, [20](#)
plotsubspace, [21](#)
posterior.cor, [22](#), [23](#)
posterior.evals, [22](#), [22](#)
posterior.mode, [23](#)
prunePed, [24](#)
Ptensor, [25](#)

rbv, [25](#)
rgl, [21](#)
rIW, [20](#), [26](#)
riwish, [27](#)
rwish, [27](#)
rwishart, [27](#)

sm2asreml, [28](#)
SShorns, [29](#)

Tri2M, [29](#)