

Package ‘MCMCglmm’

January 15, 2012

Title MCMC Generalised Linear Mixed Models

Version 2.15

Depends tensorA, Matrix, coda, ape, corpcor

Suggests rgl, combinat, mvtnorm, orthopolynom

Date 2011-11-10

Author Jarrod Hadfield

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

Description MCMC Generalised Linear Mixed Models

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-01-15 12:18:33

R topics documented:

MCMCglmm-package	2
at.level	3
at.set	4
BTdata	5
BTped	5
commutation	6
Ddivergence	7
Dexpressions	8
Dtensor	9
evalDtensor	10
inverseA	10
knorm	11
KPPM	12
krzanowski.test	13
kunif	14

leg	14
list2bdiag	15
MCMCglimm	16
mult.memb	19
PlodiaPO	20
PlodiaR	21
PlodiaRB	21
plot.MCMCglimm	22
plotsubspace	22
posterior.cor	23
posterior.evals	24
posterior.mode	25
predict.MCMCglimm	26
prunePed	27
Ptensor	28
rbv	28
residuals.MCMCglimm	29
rIW	30
rtnorm	31
sir	32
sm2asreml	33
spl	34
SShorns	35
summary.MCMCglimm	35
Tri2M	36
Index	38

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", "MCMCglmm") or the course notes vignette("CourseNotes", "MCMCglmm")

Author(s)

Jarrod 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

x	factor
level	factor level

Value

incidence matrix for level in x

Author(s)

Jarrod 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

x	factor
level	set of factor levels

Value

incidence matrix for the set level in x

Author(s)

Jarrod 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`*Commutation Matrix*

Description

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

Usage`commutation(m, n)`**Arguments**

`m` integer; number of rows of \mathbf{A}
`n` integer; number of columns of \mathbf{A}

Value

Commutation Matrix

Author(s)

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

ReferencesMagnus, 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)

Jarrold 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	<i>Tensor of (mixed) partial derivatives</i>
---------	--

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)

Jarrold 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	<i>Evaluates a list of (mixed) partial derivatives</i>
-------------	--

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)

Jarrod 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	<i>Inverse Relatedness Matrix and Phylogenetic Covariance Matrix</i>
----------	--

Description

Henderson (1976) and Meuwissen and Luo (1992) algorithm for inverting relatedness matrices, and Hadfield and Nakagawa (2010) 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 phylo 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 sparseMatrix
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) *Genetic Selection Evolution* 24 (4) 305:313
 Hadfield, J.D. and Nakagawa, S. (2010) *Journal of Evolutionary Biology* 23 494-508

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

DescriptionForms an $m \times m$ Kronecker Product Permutation Matrix**Usage**

KPPM(m, k)

Arguments

m	integer
k	integer

Value

Matrix

Author(s)

Jarrod 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	<i>Krzanowski's Comparison of Subspaces</i>
-----------------	---

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)

Jarrod 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

x a numeric vector at which to evaluate the polynomial
degree the degree of the polynomial. If negative, degree 0 is not evaluated
normalized 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)

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

See Also

[legendre.polynomials](#)

Examples

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

list2bdiag	<i>Forms the direct sum from a list of matrices</i>
------------	---

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)
```

MCMCglmm

*Multivariate Generalised Linear Mixed Models***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 2010). Please read the course notes: `vignette("CourseNotes", "MCMCglmm")` or the overview `vignette("Overview", "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=TRUE,
  saveZ=TRUE, saveXL=TRUE, slice=FALSE, ginverse=NULL)
```

Arguments

- | | |
|--------|--|
| fixed | <code>formula</code> for the fixed effects, multiple responses are passed as a matrix using <code>cbind</code> |
| random | <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 <code>variance.functions</code> available are <code>idv</code> , <code>idh</code> , <code>us</code> and <code>cor</code> . <code>idv</code> fits a constant variance across all components in <code>formula</code> , and <code>cor</code> fixes the variances to 1. 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 two reserved variables: <code>units</code> which index rows of the response variable and <code>trait</code> which index columns of the response variable |
| rcov | <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> |
| family | optional character vector of trait distributions. Currently, "gaussian", "poisson", "categorical", "multinomial", "ordinal", "exponential", "geometric", "cengaussian", "cenpoisson", "cenexponential", "zipoisson", "zapoisson", |

"ztpoisson", "hupoisson" and "zibinomial" are supported, where the prefix "cen" means censored, the prefix "zi" means zero inflated, the prefix "za" means zero altered, the prefix "zt" means zero truncated and the prefix "hu" means hurdle. If NULL, data needs to contain a family column.

mev	optional vector of measurement error variances for each data point for random effect meta-analysis.
data	data.frame
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). B is a list containing the expected value (μ) and a (co)variance matrix (V) representing the strength of belief: the defaults are $B\mu=0$ and $BV=I*1e+10$, where where I is an identity matrix of appropriate dimension. The priors for the variance structures (R and G) are lists with the expected (co)variances (V) and degree of belief parameter (ν) for the inverse-Wishart, and also the mean vector ($\alpha.\mu$) and covariance matrix ($\alpha.V$) for the redundant working parameters. The defaults are $\nu=0$, $V=1$, $\alpha.\mu=0$, and $\alpha.V=0$. When $\alpha.V$ is non-zero, parameter expanded algorithms are used.
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. This argument is retained for back compatability - see ginverse argument for a more general formulation.
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
saveXL	logical: save structural parameter design matrix
slice	logical: should slice sampling be used? Only applicable for binary traits with independent residuals
ginverse	a list of sparse inverse matrices (\mathbf{A}^{-1}) that are proportional to the covariance structure of the random effects. The names of the matrices should correspond to columns in data that are associated with the random term. All levels of the random term should appear as rownames for the matrices.

Value

So1	Posterior Distribution of MME solutions, including fixed effects
VCV	Posterior Distribution of (co)variance matrices
CP	Posterior Distribution of cut-points from an ordinal model
Liab	Posterior Distribution of latent variables
Fixed	list: fixed formula and number of fixed effects
Random	list: random formula, dimensions of each covariance matrix, number of levels per covariance matrix, and term in random formula to which each covariance belongs
Residual	list: residual formula, dimensions of each covariance matrix, number of levels per covariance matrix, and term in residual formula to which each covariance belongs
Deviance	deviance $-2*\log(p(y ...))$
DIC	deviance information criterion
X	sparse fixed effect design matrix
Z	sparse random effect design matrix
XL	sparse structural parameter design matrix
error.term	residual term for each datum
family	distribution of each datum

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

- General analyses: Hadfield, J.D. (2010) Journal of Statistical Software 33 2 1-22
 Phylogenetic analyses: Hadfield, J.D. & Nakagawa, S. (2010) Journal of Evolutionary Biology 23 494-508
 Background Sorensen, D. & Gianola, D. (2002) Springer

See Also[mcmc](#)**Examples**

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

data(PlodiaP0)
model1<-MCMCgImm(P0~1, random=~FSfamily, data=PlodiaP0, verbose=FALSE)
summary(model1)

# 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, taxon=row.names(phenotype))

Ainv<-inverseA(bird.families)$Ainv

# inverse matrix of shared phylogenetic history

prior<-list(R=list(V=1, nu=0.002), G=list(G1=list(V=1, nu=0.002)))

model2<-MCMCgImm(phenotype~1, random=~taxon, ginverse=list(taxon=Ainv),
data=test.data, prior=prior, verbose=FALSE)

plot(model2$VCV)
```

mult.memb

*Design Matrices for Multiple Membership Models***Description**

Forms design matrices for multiple membership models

Usage

```
mult.memb(formula)
```

Arguments

```
formula      formula
```

Details

Currently `mult.memb` can only usefully be used inside an `idv` variance function. The formula usually contains several factors that have the same factor levels.

Value

design matrix

Author(s)

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

Examples

```
fac1<-factor(sample(letters[1:3], 5, TRUE), levels=letters[1:3])
fac2<-factor(sample(letters[1:3], 5, TRUE), levels=letters[1:3])
cbind(fac1, fac2)
mult.memb(~fac1+fac2)
```

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	<i>Resistance of Indian meal moth caterpillars to the granulosis virus PiGV.</i>
---------	--

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	<i>Resistance (as a binary trait) of Indian meal moth caterpillars to the granulosis virus PiGV.</i>
----------	--

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](#)

plot.MCMCglmm	<i>Plots MCMC chains from MCMCglmm using plot.mcmc</i>
---------------	--

Description

plot method for class "MCMCglmm".

Usage

```
## S3 method for class 'MCMCglmm'
plot(x, random=FALSE, ...)
```

Arguments

x	an object of class "MCMCglmm"
random	logical; should saved random effects be plotted
...	Further arguments to be passed

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

See Also

[plot.mcmc](#), [MCMCglmm](#)

plotsubspace	<i>Plots covariance matrices</i>
--------------	----------------------------------

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)

Jarrod 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)

Jarrold 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)

Jarrold 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)

Jarrold 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")
```

predict.MCMCglmm *Predict method for GLMMs fitted with MCMCglmm*

Description

Predicted values for GLMMs fitted with MCMCglmm

Usage

```
## S3 method for class 'MCMCglmm'
predict(object, newdata=NULL, marginal=object$Random$formula,
        type="response", interval="none", level=0.95, ...)
```

Arguments

object	an object of class "MCMCglmm"
newdata	An optional data frame in which to look for variables with which to predict
marginal	formula defining random effects to be marginalised
type	character; either "terms" (link scale) or "response" (data scale)
interval	character; either "none", "confidence" or "prediction"
level	A numeric scalar in the interval (0,1) giving the target probability content of the intervals.
...	Further arguments to be passed

Value

Expectation and credible interval

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[MCMCglmm](#)

prunePed	<i>Pedigree pruning</i>
----------	-------------------------

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<-matrix(rnorm(n), n/2, 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 phylo object.
G	(co)variance matrix
nodes	effects for pedigree/phylogeny nodes to be returned. The default, nodes="ALL" returns effects for all individuals in a pedigree or nodes in a phylogeny (including ancestral nodes). For phylogenies nodes="TIPS" returns 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 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))
```

residuals.MCMCglmm *Residuals form a GLMM fitted with MCMCglmm*

Description

residuals method for class "MCMCglmm".

Usage

```
## S3 method for class 'MCMCglmm'
residuals(object, type = c("deviance", "pearson", "working",
                           "response", "partial"), ...)
```

Arguments

object	an object of class "MCMCglmm"
type	the type of residuals which should be returned. The alternatives are: "deviance" (default), "pearson", "working", "response", and "partial".
...	Further arguments to be passed

Value

vector of residuals

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[residuals](#), [MCMCglmm](#)

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 nu 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 fix!=NULL, V ₂₂ 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 $n \times \text{length}(V)$

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

Author(s)

Jarrod 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)
```

 rtnorm

Random Generation from a Truncated Normal Distribution

Description

Samples from the Truncated Normal Distribution

Usage

```
rtnorm(n = 1, mean = 0, sd = 1, lower = -Inf, upper = Inf)
```

Arguments

n	integer: number of samples to be drawn
mean	vector of means
sd	vector of standard deviations
lower	left truncation point
upper	right truncation point

Value

vector

Author(s)

Jarrold Hadfield j.hadfield@ed.ac.uk

References

Robert, C.P. (1995) *Statistics & Computing* 5 121-125

See Also

[rtnorm](#)

Examples

```
hist(rtnorm(100, lower=-1, upper=1))
```

sir	<i>Design Matrix for Simultaneous and Recursive Relationships between Responses</i>
-----	---

Description

Forms design matrix for simultaneous and recursive relationships between responses

Usage

```
sir(formula1=NULL, formula2=NULL)
```

Arguments

formula1	formula
formula2	formula

Value

design matrix

Author(s)

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

Examples

```
fac1<-factor(sample(letters[1:3], 5, TRUE), levels=letters[1:3])
fac2<-factor(sample(letters[1:3], 5, TRUE), levels=letters[1:3])
cbind(fac1, fac2)
sir(~fac1, ~fac2)
```

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=NULL, rownames=NULL)
```

Arguments

A	sparseMatrix
rownames	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)
```

spl *Orthogonal Spline Design Matrix*

Description

Orthogonal Spline Design Matrix

Usage

```
spl(x, k=10, knots=NULL, type="LRTP")
```

Arguments

x	a numeric covariate
k	integer, defines knot points at the $1:k/(k+1)$ quantiles of x
knots	vector of knot points
type	type of spline - currently only low-rank thin-plate ("LRTP") are implemented

Value

Design matrix post-multiplied by the inverse square root of the penalty matrix

Author(s)

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

Examples

```
## Not run:
x<-rnorm(100)
y<-x^2+cos(x)-x+0.2*x^3+rnorm(100)
plot(y~x)
lines((x^2+cos(x)-x+0.2*x^3)[order(x)]~sort(x))

dat<-data.frame(y=y, x=x)

m1<-MCMCg1mm(y~x, random=~idv(spl(x)), data=dat, pr=TRUE, saveX=TRUE, saveZ=TRUE, verbose=FALSE) # penalised smoo
m2<-MCMCg1mm(y~x+spl(x),data=dat, saveX=TRUE, verbose=FALSE) # non-penalised

pred1<-(cBind(m1$X,m1$Z))
pred2<-(cBind(m2$X))

lines(pred1[order(x)]~sort(x), col="red")
lines(pred2[order(x)]~sort(x), col="green")

m1$DIC-mean(m1$Deviance) # effective number of parameters < 13
m2$DIC-mean(m2$Deviance) # effective number of parameters ~ 13

## End(Not run)
```

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

summary.MCMCglmm

Summarising GLMM Fits from MCMCglmm

Description

summary method for class "MCMCglmm". The returned object is suitable for printing with the print.summary.MCMCglmm method.

Usage

```
## S3 method for class 'MCMCglmm'
summary(object, random=FALSE, ...)
```

Arguments

object	an object of class "MCMCglmm"
random	logical: should the random effects be summarised
...	Further arguments to be passed

Value

DIC	Deviance Information Criterion
fixed.formula	model formula for the fixed terms
random.formula	model formula for the random terms
residual.formula	model formula for the residual terms
solutions	posterior mean, 95% HPD interval, MCMC p-values and effective sample size of fixed (and random) effects
Gcovariances	posterior mean, 95% HPD interval and effective sample size of random effect (co)variance components
Rcovariances	posterior mean, 95% HPD interval and effective sample size of residual (co)variance components
cutpoints	posterior mean, 95% HPD interval and effective sample size of cut-points from an ordinal model
csats	chain length, burn-in and thinning interval
Gterms	indexes random effect (co)variances by the component terms defined in the random formula

Author(s)

Jarrod Hadfield j.hadfield@ed.ac.uk

See Also

[MCMCglmm](#)

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

x	Matrix or vector
lower.tri	If x is a matrix then the lower triangle (TRUE) or upper triangle FALSE elements (including diagonal elements) are returned. If x is a vector a matrix is formed under the assumption that x are the lower triangle (TRUE) or upper triangle (FALSE) elements.
reverse	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, 6
 - Dtensor, 9
 - evalDtensor, 10
 - inverseA, 10
 - KPPM, 12
 - Ptensor, 28
 - *Topic **datasets**
 - BTdata, 5
 - BTped, 5
 - PlodiaPO, 20
 - PlodiaR, 21
 - PlodiaRB, 21
 - SShorns, 35
 - *Topic **distribution**
 - knorm, 11
 - kunif, 14
 - posterior.cor, 23
 - posterior.ivals, 24
 - posterior.mode, 25
 - rbv, 28
 - rIW, 30
 - rtnorm, 31
 - *Topic **hplot**
 - plot.MCMCg1mm, 22
 - plotsubspace, 22
 - *Topic **manip**
 - at.level, 3
 - at.set, 4
 - leg, 14
 - list2bdiag, 15
 - mult.memb, 19
 - prunePed, 27
 - sir, 32
 - sm2asreml, 33
 - spl, 34
 - Tri2M, 36
 - *Topic **math**
 - Dtensor, 9
 - evalDtensor, 10
 - *Topic **models**
 - MCMCg1mm, 16
 - predict.MCMCg1mm, 26
 - residuals.MCMCg1mm, 29
 - summary.MCMCg1mm, 35
 - *Topic **multivariate**
 - Ddivergence, 7
 - krzanowski.test, 13
 - *Topic **symbolmath**
 - Dexpressions, 8
- at.level, 3, 4
at.set, 3, 4
- BTdata, 5
BTped, 5, 5, 6
- commutation, 6
- D, 9, 10
Ddivergence, 7
density, 25
Dexpressions, 8, 9
dnorm, 12
Dtensor, 8, 9, 10
dunif, 14
- evalDtensor, 9, 10
- formula, 16
- inverseA, 10
- knorm, 11
KPPM, 12
krzanowski.test, 13
kunif, 14
- leg, 14
legendre.polynomials, 15

list2bdiag, 15

mcmc, 19

MCMCglmm, 16, 22, 26, 30, 31, 36

MCMCglmm-package, 2

mult.memb, 19

PlodiaPO, 20, 21, 22

PlodiaR, 20, 21, 22

PlodiaRB, 20, 21, 21

plot.MCMCglmm, 22

plotsubspace, 22

posterior.cor, 23, 24

posterior.evals, 24, 24

posterior.mode, 25

predict.MCMCglmm, 26

print.MCMCglmm (summary.MCMCglmm), 35

print.summary.MCMCglmm
(summary.MCMCglmm), 35

prunePed, 27

Ptensor, 28

rbv, 28

residuals, 30

residuals.MCMCglmm, 29

rgl, 23

rIW, 30

riwish, 31

rtnorm, 31, 32

rwish, 31

rwishart, 31

sir, 32

sm2asrem1, 33

spl, 34

SShorns, 35

summary.MCMCglmm, 35

Tri2M, 36