Package 'rrBLUP'

January 28, 2018

Title Ridge Regression and Other Kernels for Genomic Selection

Version 4.6

Author Jeffrey Endelman

Maintainer Jeffrey Endelman <endelman@wisc.edu>

Depends R (>= 2.14)

Imports stats, graphics, grDevices

Suggests parallel

Description Software for genomic prediction with the RR-BLUP mixed model (Endelman 2011, <doi:10.3835/plantgenome2011.08.0024>). One application is to estimate marker effects by ridge regression; alternatively, BLUPs can be calculated based on an additive relationship matrix or a Gaussian kernel.

License GPL-3

URL http://potatobreeding.cals.wisc.edu/software

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-28 22:33:52 UTC

R topics documented:

rrBLUP-package *Ridge regression and other kernels for genomic selection*

Description

This package has been developed primarily for genomic prediction with mixed models (but it can also do genome-wide association mapping with GWAS). The heart of the package is the function mixed.solve, which is a general-purpose solver for mixed models with a single variance component other than the error. Genomic predictions can be made by estimating marker effects (RR-BLUP) or by estimating line effects (G-BLUP). In Endelman (2011) I made the poor choice of using the letter G to denotype the genotype or marker data. To be consistent with Endelman (2011) I have retained this notation in kinship.BLUP. However, that function has now been superseded by kin.blup and A.mat, the latter being a utility for estimating the additive relationship matrix (A) from markers. In these newer functions I adopt the usual convention that G is the genetic covariance (not the marker data), which is also consistent with the notation in Endelman and Jannink (2012).

Vignettes illustrating some of the features of this package can be found at http://potatobreeding.cals.wisc.edu/software.

References

Endelman, J.B. 2011. Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome 4:250-255. doi: 10.3835/plantgenome2011.08.0024

Endelman, J.B., and J.-L. Jannink. 2012. Shrinkage estimation of the realized relationship matrix. G3:Genes, Genomes, Genetics 2:1405-1413. doi: 10.1534/g3.112.004259

A.mat *Additive relationship matrix*

Description

Calculates the realized additive relationship matrix.

Usage

```
A.mat(X,min.MAF=NULL,max.missing=NULL,impute.method="mean",tol=0.02,
      n.core=1,shrink=FALSE,return.imputed=FALSE)
```

Arguments

X	Matrix ($n \times m$) of unphased genotypes for n lines and m biallelic markers, coded as {-1,0,1}. Fractional (imputed) and missing values (NA) are allowed.
min.MAF	Minimum minor allele frequency. The A matrix is not sensitive to rare alleles, so by default only monomorphic markers are removed.

max.missing	Maximum proportion of missing data; default removes completely missing markers.
impute.method	There are two options. The default is "mean", which imputes with the mean for each marker. The "EM" option imputes with an EM algorithm (see details).
tol	Specifies the convergence criterion for the EM algorithm (see details).
n.core	Specifies the number of cores to use for parallel execution of the EM algorithm (use only at UNIX command line).
shrink	Set shrink=FALSE to disable shrinkage estimation. See Details for how to enable shrinkage estimation.
return.imputed	When TRUE, the imputed marker matrix is returned.

Details

At high marker density, the relationship matrix is estimated as $A = WW'/c$, where $W_{ik} = X_{ik} + 1 - 2p_k$ and p_k is the frequency of the 1 allele at marker k. By using a normalization constant of $c = 2\sum_k p_k(1 - p_k)$, the mean of the diagonal elements is $1 + f$ (Endelman and Jannink 2012).

The EM imputation algorithm is based on the multivariate normal distribution and was designed for use with GBS (genotyping-by-sequencing) markers, which tend to be high density but with lots of missing data. Details are given in Poland et al. (2012). The EM algorithm stops at iteration t when the RMS error = $n^{-1}\|A_t - A_{t-1}\|_2$ < tol.

Shrinkage estimation can improve the accuracy of genome-wide marker-assisted selection, particularly at low marker density (Endelman and Jannink 2012). The shrinkage intensity ranges from 0 (no shrinkage) to 1 ($A = (1 + f)I$). Two algorithms for estimating the shrinkage intensity are available. The first is the method described in Endelman and Jannink (2012) and is specified by `shrink=list(method="EJ")`. The second involves designating a random sample of the markers as simulated QTL and then regressing the A matrix based on the QTL against the A matrix based on the remaining markers (Yang et al. 2010; Mueller et al. 2015). The regression method is specified by `shrink=list(method="REG",n.qtl=100,n.iter=5)`, where the parameters `n.qtl` and `n.iter` can be varied to adjust the number of simulated QTL and number of iterations, respectively.

The shrinkage and EM-imputation options are designed for opposite scenarios (low vs. high density) and cannot be used simultaneously. When the EM algorithm is used, the imputed alleles can lie outside the interval [-1,1]. Polymorphic markers that do not meet the min.MAF and max.missing criteria are not imputed.

Value

If return.imputed = FALSE, the $n \times n$ additive relationship matrix is returned.

If return.imputed = TRUE, the function returns a list containing

$A the A matrix

$imputed the imputed marker matrix

References

Endelman, J.B., and J.-L. Jannink. 2012. Shrinkage estimation of the realized relationship matrix. G3:Genes, Genomes, Genetics. 2:1405-1413. doi: 10.1534/g3.112.004259

Mueller et al. 2015. Shrinkage estimation of the genomic relationship matrix can improve genomic estimated breeding values in the training set. Theor Appl Genet doi: 10.1007/s00122-015-2464-6

Poland, J., J. Endelman et al. 2012. Genomic selection in wheat breeding using genotyping-by-sequencing. Plant Genome 5:103-113. doi: 10.3835/plantgenome2012.06.0006

Yang et al. 2010. Common SNPs explain a large proportion of the heritability for human height. Nat. Genetics 42:565-569.

Examples

```
#random population of 200 lines with 1000 markers
X <- matrix(rep(0,200*1000),200,1000)
for (i in 1:200) {
  X[i,] <- ifelse(runif(1000)<0.5,-1,1)
}

A <- A.mat(X)
```

GWAS *Genome-wide association analysis*

Description

Performs genome-wide association analysis based on the mixed model (Yu et al. 2006):

$$y = X\beta + Zg + S\tau + \varepsilon$$

where β is a vector of fixed effects that can model both environmental factors and population structure. The variable g models the genetic background of each line as a random effect with $Var[g] = K\sigma^2$. The variable τ models the additive SNP effect as a fixed effect. The residual variance is $Var[\varepsilon] = I\sigma_e^2$.

Usage

```
GWAS(pheno, geno, fixed=NULL, K=NULL, n.PC=0,
     min.MAF=0.05, n.core=1, P3D=TRUE, plot=TRUE)
```

Arguments

pheno	Data frame where the first column is the line name (gid). The remaining columns can be either a phenotype or the levels of a fixed effect. Any column not designated as a fixed effect is assumed to be a phenotype.
geno	Data frame with the marker names in the first column. The second and third columns contain the chromosome and map position (either bp or cM), respectively, which are used only when plot=TRUE to make Manhattan plots. If the markers are unmapped, just use a placeholder for those two columns. Columns 4 and higher contain the marker scores for each line, coded as {-1,0,1} = {aa,Aa,AA}. Fractional (imputed) and missing (NA) values are allowed. The column names must match the line names in the "pheno" data frame.

fixed	An array of strings containing the names of the columns that should be included as (categorical) fixed effects in the mixed model.
K	Kinship matrix for the covariance between lines due to a polygenic effect. If not passed, it is calculated from the markers using A.mat.
n.PC	Number of principal components to include as fixed effects. Default is 0 (equals K model).
min.MAF	Specifies the minimum minor allele frequency (MAF). If a marker has a MAF less than min.MAF, it is assigned a zero score.
n.core	Setting n.core > 1 will enable parallel execution on a machine with multiple cores (use only at UNIX command line).
P3D	When P3D=TRUE, variance components are estimated by REML only once, without any markers in the model. When P3D=FALSE, variance components are estimated by REML for each marker separately.
plot	When plot=TRUE, qq and Manhattan plots are generated.

Details

For unbalanced designs where phenotypes come from different environments, the environment mean can be modeled using the fixed option (e.g., fixed="env" if the column in the pheno data.frame is called "env"). When principal components are included (P+K model), the loadings are determined from an eigenvalue decomposition of the K matrix.

The terminology "P3D" (population parameters previously determined) was introduced by Zhang et al. (2010). When P3D=FALSE, this function is equivalent to EMMA with REML (Kang et al. 2008). When P3D=TRUE, it is equivalent to EMMAX (Kang et al. 2010). The P3D=TRUE option is faster but can underestimate significance compared to P3D=FALSE.

The dashed line in the Manhattan plots corresponds to an FDR rate of 0.05 and is calculated using the qvalue package (Storey and Tibshirani 2003). The p-value corresponding to a q-value of 0.05 is determined by interpolation. When there are no q-values less than 0.05, the dashed line is omitted.

Value

Returns a data frame where the first three columns are the marker name, chromosome, and position, and subsequent columns are the marker scores $(-log_{10}p)$ for the traits.

References

Kang et al. 2008. Efficient control of population structure in model organism association mapping. Genetics 178:1709-1723.

Kang et al. 2010. Variance component model to account for sample structure in genome-wide association studies. Nat. Genet. 42:348-354.

Storey and Tibshirani. 2003. Statistical significance for genome-wide studies. PNAS 100:9440-9445.

Yu et al. 2006. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Genetics 38:203-208.

Zhang et al. 2010. Mixed linear model approach adapted for genome-wide association studies. Nat. Genet. 42:355-360.

Examples

```
#random population of 200 lines with 1000 markers
M <- matrix(rep(0,200*1000),1000,200)
for (i in 1:200) {
  M[,i] <- ifelse(runif(1000)<0.5,-1,1)
}
colnames(M) <- 1:200
geno <- data.frame(marker=1:1000,chrom=rep(1,1000),pos=1:1000,M,check.names=FALSE)

QTL <- 100*(1:5) #pick 5 QTL
u <- rep(0,1000) #marker effects
u[QTL] <- 1
g <- as.vector(crossprod(M,u))
h2 <- 0.5
y <- g + rnorm(200,mean=0,sd=sqrt((1-h2)/h2*var(g)))

pheno <- data.frame(line=1:200,y=y)
scores <- GWAS(pheno,geno,plot=FALSE)
```

---

kin.blup *Genotypic value prediction based on kinship*

Description

Genotypic value prediction by G-BLUP, where the genotypic covariance G can be additive or based on a Gaussian kernel.

Usage

```
kin.blup(data,geno,pheno,GAUSS=FALSE,K=NULL,fixed=NULL,covariate=NULL,
         PEV=FALSE,n.core=1,theta.seq=NULL)
```

Arguments

data	Data frame with columns for the phenotype, the genotype identifier, and any environmental variables.
geno	Character string for the name of the column in the data frame that contains the genotype identifier.
pheno	Character string for the name of the column in the data frame that contains the phenotype.
GAUSS	To model genetic covariance with a Gaussian kernel, set GAUSS=TRUE and pass the Euclidean distance for K (see below).
K	There are three options for specifying kinship: (1) If K=NULL, genotypes are assumed to be independent ($G = I\ V_g$). (2) For breeding value prediction, set GAUSS=FALSE and use an additive relationship matrix for K to create the model ($G = K\ V_g$). (3) For the Gaussian kernel, set GAUSS=TRUE and pass the Euclidean distance matrix for K to create the model $G_{ij} = e^{-(K_{ij}/\theta)^2}\ V_g$.

`fixed`	An array of strings containing the names of columns that should be included as (categorical) fixed effects in the mixed model.
`covariate`	An array of strings containing the names of columns that should be included as covariates in the mixed model.
`PEV`	When PEV=TRUE, the function returns the prediction error variance for the genotypic values ($PEV_i = Var[g_i^* - g_i]$).
`n.core`	Specifies the number of cores to use for parallel execution of the Gaussian kernel method (use only at UNIX command line).
`theta.seq`	The scale parameter for the Gaussian kernel is set by maximizing the restricted log-likelihood over a grid of values. By default, the grid is constructed by dividing the interval (0,max(K)] into 10 points. Passing a numeric array to this variable (theta.seq = "theta sequence") will specify a different set of grid points (e.g., for large problems you might want fewer than 10).

Details

This function is a wrapper for `mixed.solve` and thus solves mixed models of the form:

$$y = X\beta + [Z\ 0]g + \varepsilon$$

where β is a vector of fixed effects, g is a vector of random genotypic values with covariance $G = Var[g]$, and the residuals follow $Var[\varepsilon_i] = R_i\sigma_e^2$, with $R_i = 1$ by default. The design matrix for the genetic values has been partitioned to illustrate that not all lines need phenotypes (i.e., for genomic selection). Unlike `mixed.solve`, this function does not return estimates of the fixed effects, only the BLUP solution for the genotypic values. It was designed to replace `kinship.BLUP` and to relieve the user of having to explicitly construct design matrices. Variance components are estimated by REML and BLUP values are returned for every entry in K, regardless of whether it has been phenotyped. The rownames of K must match the genotype labels in the data frame for phenotyped lines; missing phenotypes (NA) are simply omitted.

Unlike its predecessor, this function does not handle marker data directly. For breeding value prediction, the user must supply a relationship matrix, which can be calculated from markers with `A.mat`. For Gaussian kernel predictions, pass the Euclidean distance matrix for K, which can be calculated with `dist`.

In the terminology of mixed models, both the "fixed" and "covariate" variables are fixed effects (β in the above equation): the former are treated as factors with distinct levels while the latter are continuous with one coefficient per variable. The population mean is automatically included as a fixed effect.

The prediction error variance (PEV) is the square of the SE of the BLUPs (see `mixed.solve`) and can be used to estimate the expected accuracy of BLUP predictions according to $r_i^2 = 1 - \frac{PEV_i}{V_g K_{ii}}$.

Value

The function always returns

$Vg REML estimate of the genetic variance

$Ve REML estimate of the error variance

$g BLUP solution for the genetic values

$resid residuals

$pred predicted genetic values, averaged over the fixed effects

If PEV = TRUE, the list also includes

$PEV Prediction error variance for the genetic values

If GAUSS = TRUE, the list also includes

$profile the log-likelihood profile for the scale parameter in the Gaussian kernel

References

Endelman, J.B. 2011. Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome 4:250-255. doi: 10.3835/plantgenome2011.08.0024

Examples

```
#random population of 200 lines with 1000 markers
M <- matrix(rep(0,200*1000),200,1000)
for (i in 1:200) {
  M[i,] <- ifelse(runif(1000)<0.5,-1,1)
}
rownames(M) <- 1:200
A <- A.mat(M)

#random phenotypes
u <- rnorm(1000)
g <- as.vector(crossprod(t(M),u))
h2 <- 0.5  #heritability
y <- g + rnorm(200,mean=0,sd=sqrt((1-h2)/h2*var(g)))

data <- data.frame(y=y,gid=1:200)

#predict breeding values
ans <- kin.blup(data=data,geno="gid",pheno="y",K=A)
accuracy <- cor(g,ans$g)
```

`kinship.BLUP` *Genomic prediction by kinship-BLUP (deprecated)*

Description

***This function has been superseded by `kin.blup`; please refer to its help page.

Usage

```
kinship.BLUP(y, G.train, G.pred=NULL, X=NULL, Z.train=NULL,
     K.method="RR", n.profile=10, mixed.method="REML", n.core=1)
```

Arguments

y	Vector ($n.obs \times 1$) of observations. Missing values (NA) are omitted.
G.train	Matrix ($n.train \times m$) of unphased genotypes for the training population: $n.train$ lines with m bi-allelic markers. Genotypes should be coded as {-1,0,1}; fractional (imputed) and missing (NA) alleles are allowed.
G.pred	Matrix ($n.pred \times m$) of unphased genotypes for the prediction population: $n.pred$ lines with m bi-allelic markers. Genotypes should be coded as {-1,0,1}; fractional (imputed) and missing (NA) alleles are allowed.
X	Design matrix ($n.obs \times p$) of fixed effects. If not passed, a vector of 1's is used to model the intercept.
Z.train	0-1 matrix ($n.obs \times n.train$) relating observations to lines in the training set. If not passed the identity matrix is used.
K.method	"RR" (default) is ridge regression, for which K is the realized additive relationship matrix computed with A.mat. The option "GAUSS" is a Gaussian kernel ($K = e^{-D^2/\theta^2}$) and "EXP" is an exponential kernel ($K = e^{-D/\theta}$), where Euclidean distances D are computed with dist.
n.profile	For K.method = "GAUSS" or "EXP", the number of points to use in the log-likelihood profile for the scale parameter θ.
mixed.method	Either "REML" (default) or "ML".
n.core	Setting n.core > 1 will enable parallel execution of the Gaussian kernel computation (use only at UNIX command line).

Value

$g.train BLUP solution for the training set

$g.pred BLUP solution for the prediction set (when G.pred != NULL)

$beta ML estimate of fixed effects

For GAUSS or EXP, function also returns

$profile log-likelihood profile for the scale parameter

References

Endelman, J.B. 2011. Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome 4:250-255.

Examples

```
#random population of 200 lines with 1000 markers
G <- matrix(rep(0,200*1000),200,1000)
for (i in 1:200) {
  G[i,] <- ifelse(runif(1000)<0.5,-1,1)
}

#random phenotypes
g <- as.vector(crossprod(t(G),rnorm(1000)))
```

```
h2 <- 0.5
y <- g + rnorm(200,mean=0,sd=sqrt((1-h2)/h2*var(g)))

#split in half for training and prediction
train <- 1:100
pred <- 101:200
ans <- kinship.BLUP(y=y[train],G.train=G[train,],G.pred=G[pred,],K.method="GAUSS")

#correlation accuracy
r.gy <- cor(ans$g.pred,y[pred])
```

`mixed.solve` *Mixed-model solver*

Description

Calculates maximum-likelihood (ML/REML) solutions for mixed models of the form

$$y = X\beta + Zu + \varepsilon$$

where β is a vector of fixed effects and u is a vector of random effects with $Var[u] = K\sigma_u^2$. The residual variance is $Var[\varepsilon] = I\sigma_e^2$. This class of mixed models, in which there is a single variance component other than the residual error, has a close relationship with ridge regression (ridge parameter $\lambda = \sigma_e^2/\sigma_u^2$).

Usage

```
mixed.solve(y, Z=NULL, K=NULL, X=NULL, method="REML",
        bounds=c(1e-09, 1e+09), SE=FALSE, return.Hinv=FALSE)
```

Arguments

`y`	Vector ($n \times 1$) of observations. Missing values (NA) are omitted, along with the corresponding rows of X and Z.
`Z`	Design matrix ($n \times m$) for the random effects. If not passed, assumed to be the identity matrix.
`K`	Covariance matrix ($m \times m$) for random effects; must be positive semi-definite. If not passed, assumed to be the identity matrix.
`X`	Design matrix ($n \times p$) for the fixed effects. If not passed, a vector of 1's is used to model the intercept. X must be full column rank (implies β is estimable).
`method`	Specifies whether the full ("ML") or restricted ("REML") maximum-likelihood method is used.
`bounds`	Array with two elements specifying the lower and upper bound for the ridge parameter.
`SE`	If TRUE, standard errors are calculated.
`return.Hinv`	If TRUE, the function returns the inverse of $H = ZKZ' + \lambda I$. This is useful for GWAS.

Details

This function can be used to predict marker effects or breeding values (see examples). The numerical method is based on the spectral decomposition of ZKZ' and $SZKZ'S$, where $S = I - X(X'X)^{-1}X'$ is the projection operator for the nullspace of X (Kang et al., 2008). This algorithm generates the inverse phenotypic covariance matrix V^{-1}, which can then be used to calculate the BLUE and BLUP solutions for the fixed and random effects, respectively, using standard formulas (Searle et al. 1992):

$$BLUE(\beta) = \beta^* = (X'V^{-1}X)^{-1}X'V^{-1}y$$

$$BLUP(u) = u^* = \sigma_u^2 KZ'V^{-1}(y - X\beta^*)$$

The standard errors are calculated as the square root of the diagonal elements of the following matrices (Searle et al. 1992):

$$Var[\beta^*] = (X'V^{-1}X)^{-1}$$

$$Var[u^* - u] = K\sigma_u^2 - \sigma_u^4 KZ'V^{-1}ZK + \sigma_u^4 KZ'V^{-1}XVar[\beta^*]X'V^{-1}ZK$$

For marker effects where K = I, the function will run faster if K is not passed than if the user passes the identity matrix.

Value

If SE=FALSE, the function returns a list containing

$Vu estimator for σ_u^2

$Ve estimator for σ_e^2

$beta BLUE($\beta$)

$u BLUP($u$)

$LL maximized log-likelihood (full or restricted, depending on method)

If SE=TRUE, the list also contains

$beta.SE standard error for β

$u.SE standard error for $u^* - u$

If return.Hinv=TRUE, the list also contains

$Hinv the inverse of H

References

Kang et al. 2008. Efficient control of population structure in model organism association mapping. Genetics 178:1709-1723.

Endelman, J.B. 2011. Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome 4:250-255.

Searle, S.R., G. Casella and C.E. McCulloch. 1992. Variance Components. John Wiley, Hoboken.

Examples

```
#random population of 200 lines with 1000 markers
M <- matrix(rep(0,200*1000),200,1000)
for (i in 1:200) {
  M[i,] <- ifelse(runif(1000)<0.5,-1,1)
}

#random phenotypes
u <- rnorm(1000)
g <- as.vector(crossprod(t(M),u))
h2 <- 0.5  #heritability
y <- g + rnorm(200,mean=0,sd=sqrt((1-h2)/h2*var(g)))

#predict marker effects
ans <- mixed.solve(y,Z=M)  #By default K = I
accuracy <- cor(u,ans$u)

#predict breeding values
ans <- mixed.solve(y,K=A.mat(M))
accuracy <- cor(g,ans$u)
```

Index