Package 'NAM'

September 29, 2018

Type Package

Title Nested Association Mapping

Version 1.6.4

Date 2018-10-01

Author Alencar Xavier, William Muir, Katy Rainey, Shizhong Xu.

Maintainer Alencar Xavier <alenxav@gmail.com>

Description Designed for association studies in nested association mapping (NAM) panels, experimental and random panels. The method is described by Xavier et al. (2015) <doi:10.1093/bioinformatics/btv448>. It includes tools for genome-wide associations of multiple populations, marker quality control, population genetics analysis, genome-wide prediction, solving mixed models and finding variance components through likelihood and Bayesian methods.

License GPL-3

Imports Rcpp,randomForest

LinkingTo Rcpp

Depends R (>= 3.2.0)

NeedsCompilation yes

Repository CRAN

Suggests knitr,rmarkdown,lme4

VignetteBuilder knitr

Date/Publication 2018-09-29 21:30:03 UTC

R topics documented:

NAM-package *Nested Association Mapping*

Description

Designed for association studies in nested association mapping (NAM) panels, also handling experimental and random panels. The method is described by Xavier et al. (2015) <doi:10.1093/bioinformatics/btv448>. It includes tools for genome-wide associations of multiple populations, marker quality control, population genetics analysis, genome-wide prediction, solving mixed models and finding variance components through likelihood and Bayesian methods.

Details

Package:	NAM
Type:	Package
Version:	1.6.4
Date:	2018-10-01
License:	GPL-3

Author(s)

Alencar Xavier, William Muir, Katy Rainey, Shizhong Xu.

Maintainer: Alencar Xavier <alenxav@gmail.com>

See Also

Package include functions to perform association analysis (gwas, gwas2, gwas3), meta-gwas (gwasGE, meta3), solve mixed models (gibbs, reml), genomic prediction (wgr, gmm, press), populations genetic analysis (Fst, Gdist, LD), data quality control (snpQC, snpH2, markov), spatial analysis (covar, SPC, SPM), build kinships (GRM, GAU, PedMat) and more.

`Dataset 1` *Tetra-seed Pods*

Description

Two soybean bi-parental crosses phenotyped for the percentage of pods containing four seeds

Usage

```
data(tpod)
```

Details

Soybean nested association panel with 2 families (fam) containing 196 indiviuals. Genotypic matrix (gen) have 376 SNP across 20 chromosome (chr). Phenotypic information (y) regards the proportion of four seed pods. Data provided by Rainey Lab for Soybean Breeding and Genetics, Purdue University.

Author(s)

Alencar Xavier and Katy Rainey

`Dataset 2` *Multi-environmental trial*

Description

Data collected from the SoyNAM population in Indiana 2013-2015.

Usage

```
data(met)
```

Details

Data provided by Rainey Lab for Soybean Breeding and Genetics, Purdue University. Genotypic matrix (Gen) have 4240 SNPs. The data frame Obs contains the soybean id (ID), environment ($Year$), field location ($Block, Row, Col$) and three phenotypes: grain yield (YLD), days to maturity (DTM) and average canopy closure (ACC).

Author(s)

Alencar Xavier, Ben Hall and Katy Rainey

References

Xavier, A., Hall, B., Hearst, A.A., Cherkauer, K.A. and Rainey, K.M., 2017. Genetic Architecture of Phenomic-Enabled Canopy Coverage in Glycine max. Genetics, 206(2), pp.1081-1089.

FST *Fixation Index*

Description

Genetic variation associated with markers distributed among subpopulations. The function generates a plot for structure diagnosis.

Usage

```
Fst(gen,fam)
```

Arguments

gen	Numeric matrix containing the genotypic data. A matrix with n rows of observations and (m) columns of molecular markers. SNPs must be coded as `0, 1, 2`, for founder homozigous, heterozygous and reference homozygous. `NA` is allowed.
fam	Numeric vector of length (n) indicating which subpopulations ($i.e.$ family) each observation comes from. `NA` is not allowed.

Details

F-statistics (Wright 1965) represent the differentiation among populations for a given locus. Weir and Cockerham (1984) provided an unbiased version for molecular analysis.

FIT is the correlation between gametes that unite to produce the individuals, relative to the gametes of the total population. FIS is the average over all subdivisions of the correlation between uniting gametes relative to those of their own subdivision. FST is the correlation between random gametes within subdivisions, relative to gametes of the total population. Neutral markers have an expected FST 0.05.

Value

List with values of FST, FIS and FIT. Unbiased F-statistics from weighted AOV (Weir and Cockerham 1984).

Author(s)

Alencar Xavier and William Muir

References

Weir, B. S., and Cockerham, C. C. (1984). Estimating F-statistics for the analysis of population structure. Evolution, 38(6), 1358-1370.

Wright, S. (1965). The interpretation of population structure by F-statistics with special regard to systems of mating. Evolution, 19(3), 395-420.

Examples

```
data(tpod)
Fstat = Fst(gen=gen,fam=fam)
plot(Fstat,chr=chr)
```

GWAS *Empirical Bayes Genome Wide Association Mapping*

Description

The gwas function calculates the likelihood ratio for each marker under the empirical Bayesian framework. The method allows analysis with multiple populations. gwas2 is computationally optimized. gwas3 was design for multiple random populations.

Usage

```
gwas(y,gen,fam=NULL,chr=NULL,window=NULL,fixed=FALSE)
gwas2(y,gen,fam=NULL,chr=NULL,fixed=FALSE,EIG=NULL,cov=NULL)
gwas3(y,gen,fam=NULL,chr=NULL,EIG=NULL,cov=NULL)
gwasGE(Phe,gen,fam,chr=NULL,cov=NULL,ge=FALSE,ammi=1)
meta3(ByEnv,ammi=1)
```

Arguments

y	Numeric vector of observations (n) describing the trait to be analyzed. NA is allowed.
gen	Numeric matrix containing the genotypic data. A matrix with n rows of observations and (m) columns of molecular markers. SNPs must be coded as 0, 1, 2, for founder homozygous, heterozygous and reference homozigous. NA is allowed.
fam	Numeric vector of length n indicating a stratification factor or which subpopulation ($e.g.$ family) that each observation comes from. Default assumes that all observations are from the same populations.
chr	Numeric vector indicating the number of markers in each chromosome. The sum of chr must be equal to the number of columns in gen. Default assumes that all markers are from the same chromosome.
window	Numeric. If specified, genetic distance between markers is used for moving window strategy. Window must be specified in Morgans ($e.g.$ 0.05 would represent 5cM). Genetic distance is calculated assuming that individuals are RILs.
fixed	Logical. If TRUE, markers are treated as fixed effect and hence, evaluated through Wald statistics. If markers are specief as fixed, the argument 'window' is not applicable.
EIG	Output of the R function 'eigen'. It is used for user-defined kinship matrix.
cov	Numeric vector of length n to be used as covariate in the association analysis.

Phe	Numeric matrix of observations ($n * e$) where rows represent genotypes and columns represent environments. NA is allowed.
ge	Logical. If TRUE, meta-analysis (function gwasGE) will be done for the GxE interactions term only. If FALSE, variance components will be computed for three terms: genotype, environment and interaction.
ammi	Integer. It indicates the number of principal components used to represent GxE interactions through additive main-effects and multiplicative interaction (AMMI).
ByEnv	List of objected output from gwas3 to perform meta-analysis.

Details

Empirical Bayes model (Wang 2016) with a special incidence matrix is recreated to optimize the information provided by the subpopulations. Each locus is recoded as a vector with length f equal to number of subpopulations, or NAM families, as the interaction locus by family. For example, a locus heterozigous from an individual from subpopulation 2 is coded as [1, 0, 1 , ... ,f], a locus homozigous for the reference allele from any subpopulation is coded as [2, 0, 0, ... , f] and a locus homozigous for the founder allele from an individual from subpopulation 1 is coded as [0, 2, 0, ... ,f]. The base model for genome scanning is described by:

$$y = Xb + Zu + g + e$$

That includes the fixed effect (Xb), the marker (Zu), the polygene (g) and the residuals (e). If the $window$ term is specified, the model for genome scanning is expanded as follows:

$$y = Xb + Zu[k-1] + Zu + Zu[k+1] + g - g[k] + e$$

This model includes three extra terms: the left side genome ($Zu[k-1]$) and the right side genome ($Zu[k+1]$), also subtracting the window polygene ($-g[k]$). Windows are based on genetic distance, which is computed using Kosambi map function. The recombination rate is estimated under the assuption markers are ordered and that genotypes are recombinant inbred lines.

The polygenic term is calculated only once (Zhang et al 2010) using eigendecomposition with a GEMMA-like algorithm (Zhou ans Stephens 2012). Efficient inversion of capacitance matrix is obtained through the Woodbury matrix identities. Models and algorithms are described with more detail by Xavier et al (2015) and Wei and Xu (2016).

In order to analyze large dateset, one can avoid memory issues by using the function $gwas2$, but that the argument 'window' is not implemented for $gwas2$. This function also allows used-defined kindship through the argument EIG, and the use of a numeric covariate vector through the argument cov.

When multi-environmental trials are the target of mapping, one may use the function $gwasGE$ to perform analysis by environment, followed by "meta-analysis" used to combine the results. This strategy provides an idea of the variation on QTL effect due to environment, genetic background (provided by the stratification factor) and the interaction between environment and genetics.

An alternative to this method is the mega-analysis, where one can provide the stratification factor as a combination of subpopulation and environment. Meta-analysis can be performed in a single step with function gwasGE, or users can perform multiple association analyses using gwas3 and perform meta-analysis with meta3. In gwasGE, the same genotype will often appear more than once in the

phenotypic and genotypic data, so that phenotypes are provided as a matrix. The statistical detail about the meta-analysis are available in the vignette $Backgroundfor Meta-analysis$.

The function gwas3 is an alternative for association analysis and meta-analysis, also solved in the Empirical-Bayes framework for multiple populations. Unlike gwas, gwas2 and gwasGE, this function does not set a reference allele and analysis each marker as the interaction of allele by stratification factor (ie. family or subpopulation). Therefore, gwas3 is compatible with any allele coding.

For further statistical background:

1) `system(paste('open',system.file("doc","gwa_description.pdf",package="NAM")))`

2) `system(paste('open',system.file("doc","gwa_ge_interactions.pdf",package="NAM")))`

Value

The function gwas returns a list containing the method deployed ($Method$), a summary of predicted parameters and statistical tests ($PolyTest$), estimated genetic map for NAM panels (MAP) and the marker names ($SNPs$).

Author(s)

Alencar Xavier, Tiago Pimenta, Qishan Wang and Shizhong Xu

References

Wang, Q., Wei, J., Pan, Y., & Xu, S. (2016). An efficient empirical Bayes method for genomewide association studies. Journal of Animal Breeding and Genetics, 133(4), 253-263.

Wei, J., & Xu, S. (2016). A Random Model Approach to QTL Mapping in Multi-parent Advanced Generation Inter-cross (MAGIC) Populations. Genetics, 202(2), 471-486.

Xavier, A., Xu, S., Muir, W. M., & Rainey, K. M. (2015). NAM: Association Studies in Multiple Populations. Bioinformatics, 31(23), 3862-3864.

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

Zhou, X., & Stephens, M. (2012). Genome-wide efficient mixed-model analysis for association studies. Nature genetics, 44(7), 821-824.

Examples

```
data(tpod)
gen=reference(gen)
gwa=gwas2(y=y,gen=gen,fam=fam,chr=chr,fixed=TRUE)
plot(gwa,pch=20,lwd=4)
```

GWP *Genome-wide prediction*

Description

Univariate model to find breeding values through regression with optional resampling techniques (SBMC) and polygenic term (Kernel).

Usage

```
wgr(y,X,it=1500,bi=500,th=1,bag=1,rp=FALSE,iv=FALSE,de=FALSE,
    pi=0,df=5,R2=0.5,eigK=NULL,VarK=0.95,verb=FALSE)
```

Arguments

y	Numeric vector of observations (n) describing the trait to be analyzed. NA is allowed.
X	Numeric matrix containing the genotypic data. A matrix with n rows of observations and (m) columns of molecular markers.
it	Integer. Number of iterations or samples to be generated.
bi	Integer. Burn-in, the number of iterations or samples to be discarted.
th	Integer. Thinning parameter, used to save memory by storing only one every 'th' samples.
bag	If different than one, it indicates the proportion of data to be subsampled in each Markov chain. For datasets with moderate number of observations, values of bag from 0.30 to 0.60 may speed up computation without losses in predicion properties. This argument enable users to enhance MCMC through Subsampling bootstrap Markov chain (SBMC).
rp	Logical. Use replacement for bootstrap samples when bag is different than one.
iv	Logical. Assign markers independent variance, a T prior from a mixture of normals. If true, turns the default model BLUP into BayesA.
de	Logical. Assign markers independent variance through double-exponential prior. If true, turns the default model BLUP into Bayesian LASSO. This argument overides iv.
pi	Value between 0 and 1. If greater than zero it activates variable selection, where markers have expected probability pi of having null effect.
df	Prior degrees of freedom of variance components.
R2	Expected R2, used to calculate the prior shape.
eigK	Output of function 'eigen'. Spectral decomposition of the kernel used to compute the polygenic term.
VarK	Numeric between 0 and 1. For reduction of dimensionality. Indicates the proportion of variance explained by Eigenpairs used to fit the polygenic term.
verb	Logical. If verbose is TRUE, function displays MCMC progress bar.

Details

The model for the whole-genome regression is as follows:

$$y = mu + Xb + u + e$$

where y is the response variable, mu is the intercept, X is the genotypic matrix, b is the regression coefficient or effect of an allele substitution, with d probability of being included into the model, u is the polygenic term if a kernel is used, and e is the residual term.

Users can obtain four WGR methods out of this function: BRR (pi=0,iv=F), BayesA (pi=0,iv=T), BayesB (pi=0.95,iv=T), BayesC (pi=0.95,iv=F) and Bayesian LASSO or BayesL (pi=0,de=T). Theoretical basis of each model is described by de los Campos et al. (2013).

Gibbs sampler that updates regression coefficients is adapted from GSRU algorithm (Legarra and Misztal 2008). The variable selection works through the unconditional prior algorithm proposed by Kuo and Mallick (1998). Prior shape estimates are computed as Sb = R2*df*var(y)/MSx and Se = (1-R2)*df*var(y). The polygenic term is solved by Bayesian algorithm of reproducing kernel Hilbert Spaces proposed by de los Campos et al. (2010).

Value

The function wgr returns a list with expected value from the marker effect (b), probability of marker being in the model (d), regression coefficient (g), variance of each marker (Vb), the intercept (mu), the polygene (u) and polygenic variance (Vk), residual variance (Ve) and the fitted value (hat).

Author(s)

Alencar Xavier

References

de los Campos, G., Hickey, J. M., Pong-Wong, R., Daetwyler, H. D., and Calus, M. P. (2013). Whole-genome regression and prediction methods applied to plant and animal breeding. Genetics, 193(2), 327-345.

de los Campos, G., Gianola, D., Rosa, G. J., Weigel, K. A., & Crossa, J. (2010). Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. Genetics Research, 92(04), 295-308.

Kuo, L., & Mallick, B. (1998). Variable selection for regression models. Sankhya: The Indian Journal of Statistics, Series B, 65-81.

Legarra, A., & Misztal, I. (2008). Technical note: Computing strategies in genome-wide selection. Journal of dairy science, 91(1), 360-366.

Examples

```
# Load data
data(tpod)

# BLUP
BRR = wgr(y,gen,iv=FALSE,pi=0,it=200,bi=50)
```

```
cor(y,BRR$hat)

# BayesA
BayesA = wgr(y,gen,iv=TRUE,pi=0,it=200,bi=50)
cor(y,BayesA$hat)

# BayesB
BayesB = wgr(y,gen,iv=TRUE,pi=.95,it=200,bi=50)
cor(y,BayesB$hat)

# BayesC
BayesC = wgr(y,gen,iv=FALSE,pi=.95,it=200,bi=50)
cor(y,BayesC$hat)

# BayesL
BayesL = wgr(y,gen,de=TRUE,it=200,bi=50)
cor(y,BayesL$hat)
```

`Manhattan` *Manhattan plot for Association Studies*

Description

Generates a graphical visualization for the output of the function gwas/gwas2/gwasGE.

Usage

```
## S3 method for class 'NAM'
plot(x,...,alpha=0.05,colA=2,colB=4,find=NULL,FDR=NULL,gtz=FALSE,phys=NULL)
```

Arguments

`x`	Output of the gwas/gwas2/gwas3 function.
`...`	Further arguments passed to or from other methods.
`alpha`	Numberic. Significance threshold to display in the Manhattan plot.
`colA`	Color of odd chromosomes in the Manhattan plot.
`colB`	Color of even chromosomes in the Manhattan plot.
`find`	Integer. If provided, you can click on the specified number of hits in the Manhattan plot to obtain the name of the markers.
`FDR`	Null or numeric between zero and one. If provided, it will display the Manhattan plot with Bonferroni threshold by chromosome, adjusted for the specified false discovery rate (FDR). Thus, zero provides the Bonferroni correction.
`gtz`	Logical. If TRUE, the argument FDR will just take into account markers with p-value Greater Than Zero (GTZ).
`phys`	Numeric vector with length equal to the number of markers. If provided, the Manhattan plot is generated using the physical distance in the x axis.

Author(s)

Alencar Xavier and William Beavis

Examples

```
data(tpod)
test=gwas2(y=y,gen=gen[,1:240],fam=fam,chr=chr[1:12])
par(mfrow=c(2,1))

# Example Manhattan 1
SIGNIF = 1+(2*test$PolyTest$lrt>4.9)
plot(x=test,pch=SIGNIF+3,lwd=SIGNIF,main="Example 2")

# Example Manhattan 2
plot(x=test,main="Example 3",pch=20,lwd=2)
Kern = ksmooth(1:240,test$PolyTest$lrt,bandwidth=1)
lines(Kern,type="l",lwd=2)
```

MLM Gibbs *Bayesian Mixed Model*

Description

Mixed model solver through Bayesian Gibbs Sampling or iterative solution.

Usage

```
gibbs(y,Z=NULL,X=NULL,iK=NULL,iR=NULL,Iter=1500,Burn=500,
      Thin=2,DF=5,S=0.5,nor=TRUE,GSRU=FALSE)
ml(y,Z=NULL,X=NULL,iK=NULL,iR=NULL,DF=5,S=0.5,nor=TRUE)
gibbs2(Y,Z=NULL,X=NULL,iK=NULL,Iter=150,Burn=50,Thin=1,DF=5,S=0.5,nor=TRUE)
```

Arguments

y	Numeric vector of observations (n) describing the trait to be analyzed. NA is allowed.
Z	Right-hand side formula or list of numeric matrices (n by p) with incidence matrices for random effect. NA is not allowed.
X	Right-hand side formula or incidence matrices (n by p) for fixed effect. NA is not allowed.
iK	Numeric matrix or list of numeric matrices (p by p) corresponding to the the inverse kinship matrix of each random effect with p parameters.
iR	Numeric matrix (n by n) corresponding to the inverse residual correlation matrix.

Iter	Integer. Number of iterations or samples to be generated.
Burn	Integer. Number of iterations or samples to be discarted.
Thin	Integer. Thinning parameter, used to save memory by storing only one every 'Thin' samples.
DF	Integer. Hyperprior degrees of freedom of variance components.
S	Integer or NULL. Hyperprior shape of variance components. If NULL, the hyperprior solution for the scale parameter is calculated as proposed by de los Campos et al. (2013).
nor	Logical. If TRUE, it normilizes the response variable(s).
GSRU	Logical. If TRUE, it updates the regression coefficient using Gauss-Seidel Residual Update (Legarra and Misztal 2008). Useful for p»n, but does not work when iK or iR are provided.
Y	Numeric matrix of observations for multivariate mixed models. Each column represents a trait. NA is allowed.

Details

The general model is $y = Xb + Zu + e$, where $u = N(0, A\sigma 2a)$ and $e = N(0, R\sigma 2e)$. The function solves Gaussian mixed models in the Bayesian framework as described by Garcia-Cortes and Sorensen (1996) and Sorensen and Gianola (2002) with conjugated priors. The alternative function, "ml", finds the solution iteratively using the full-conditional expectation. The function "gibbs2" can be used for the multivariate case, check Xavier et al. (2017) for an example of multivariate mixed model using Gibbs sampling.

Value

The function gibbs returns a list with variance components distribution a posteriori (Posterior.VC) and mode estimated (VC.estimate), a list with the posterior distribution of regression coefficients (Posterior.Coef) and the posterior mean (Coef.estimate), and the fitted values using the mean (Fit.mean) of posterior coefficients.

Author(s)

Alencar Xavier

References

de los Campos, G., Hickey, J. M., Pong-Wong, R., Daetwyler, H. D., and Calus, M. P. (2013). Whole-genome regression and prediction methods applied to plant and animal breeding. Genetics, 193(2), 327-345.

Legarra, A., & Misztal, I. (2008). Technical note: Computing strategies in genome-wide selection. Journal of dairy science, 91(1), 360-366.

Garcia-Cortes, L. A., and Sorensen, D. (1996). On a multivariate implementation of the Gibbs sampler. Genetics Selection Evolution, 28(1), 121-126.

Sorensen, D., & Gianola, D. (2002). Likelihood, Bayesian, and MCMC methods in quantitative genetics. Springer Science & Business Media.

Xavier, A., Hall, B., Casteel, S., Muir, W. and Rainey, K.M. (2017). Using unsupervised learning techniques to assess interactions among complex traits in soybeans. Euphytica, 213(8), p.200.

Examples

```
data(tpod)

# Fitting GBLUP
K = GRM(gen)
iK = chol2inv(K)

# FIT
test1 = gibbs(y,iK=iK,S=1)

# PLOT
par(mfrow=c(1,3))
plot(test1$Fit.mean,y,pch=20,lwd=2,col=3,main='GBLUP')
plot(test1,col=4,lwd=2)

# Heritability
print(paste('h2 =',round(test1$VC.estimate[1]/sum(test1$VC.estimate),3)))

# Fitting RKHS
G = GAU(gen)
EIG = eigen(G,symmetric = TRUE)
ev = 20
U = EIG$vectors[,1:ev]
iV = diag(1/EIG$values[1:ev])

# FIT
test2 = gibbs(y,Z=U,iK=iV,S=1)

# PLOT
par(mfrow=c(1,3))
plot(test2$Fit.mean,y,pch=20,lwd=2,col=2,main='RKHS')
plot(test2,col=3,lwd=2)

# Heritability
print(paste('h2 =',round(test2$VC.estimate[1]/sum(test2$VC.estimate),3)))
```

MLM REML *Restricted Maximum Likelihood*

Description

Univariate REML estimators and variance components for a single random variable fitted by an EMMA-like algorithm.

Usage

```
reml(y,X=NULL,Z=NULL,K=NULL)
MCreml(y,K,X=NULL,MC=300,samp=300)
```

Arguments

y	Numeric vector of observations (n) describing the trait to be analyzed. NA is allowed.
X	Formula or incidence matrix (n by p) for fixed effect. NA is not allowed.
Z	Formula or numeric matrix (n by p) that corresponds to the incidence matrix of random effect. NA is not allowed.
K	Numeric matrix (p by p). Kinship matrix for random effect with p parameters. NA is not allowed.
MC	Number of sampling procedures to estimate variance components using MCreml.
samp	Sample size of the sampling procedure to estimate variance components using MCreml.

Details

Solve mixed models with a single random effects minizing the log restricted maximum likelihood (REML) using the EMMA algorithm (Kang et al 2008). Prediction of random coefficients for ridge-type model are performed according to VanRaden (2008), and kernel-type model via RKHS according to de los Campos et al. (2010).

If y is a matrix with multiple traits, the fuctions solves the mixed model via an ECM algorithm adapted from the EMMREML package (Akdemir and Godfrey 2014).

MCreml is based on subsampling with samp observations at a time, repeating the procedure MC times. Subsampling is a common Monte Carlo strategy to reduce the computational burden to estimate variance components in large datasets.

Value

The function reml returns a list with variance components and heritability (VC), fixed effect coefficients and standard variations (Fixed) and estimated breeding values (EBV).

Author(s)

Alencar Xavier, Tiago Pimenta and Shizhong Xu

References

Akdemir, D., and O. U. Godfrey (2014) EMMREML: Fitting Mixed Models with Known Covariance Structures. R Package Version 2.0. Available at: http://CRAN.R-project.org/package=EMMREML.

de los Campos, G., Gianola, D., Rosa, G. J., Weigel, K. A., & Crossa, J. (2010). Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. Genetics Research, 92(04), 295-308.

Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., & Eskin, E. (2008). Efficient control of population structure in model organism association mapping. Genetics, 178(3), 1709-1723.

VanRaden, P. M. (2008). Efficient methods to compute genomic predictions. Journal of dairy science, 91(11), 4414-4423.

Examples

```
# Fitting a random model
data(tpod)
FIT = reml(y=y,Z=~as.factor(fam))

# Fitting GBLUP
G = GRM(gen)
GBLUP = reml(y=y,K=G)

# GBLUP vs RRBLUP
g = tcrossprod(gen)
gblup = reml(y=y,K=g)
rrblup = reml(y=y,Z=gen)
rbind(gblup$VC,rrblup$VC)
gebv_gblup = gblup$EBV
gebv_rrblup = c(tcrossprod(t(rrblup$EBV),gen))
plot(gebv_gblup,gebv_rrblup)
```

MLM Trials *Genomic mixed model*

Description

This function was developed to solve a mixel model for multi-environmental trials and/or replicated trials when genomic is available. The model includes a semi-parametric term to account for spatial variation when the field layout of the experiment is know. Covariates (fixed effects), genetics (random effect) and spatial term are fitted all in a single step.

Usage

```
gmm(y,gen,dta=NULL,it=75,bi=25,th=1,model="BRR",...)
```

Arguments

y	Numeric vector of phenotypes of length obs (missing values are allowed). NA is allowed.
gen	Numeric matrix, with dimension n x p. Attention: Rows must be named with genotype IDs. Missing values are replaced by the SNP expectation.

dta	Data frame with *obs* number of rows containg the genotype ID, spatial information and any other set covariates. Make sure to add a column called "ID" (with capital letters) informing the genotype ID with some match in the object gen. For the spatial adjustment, it is necessary to add three numeric columns (not factors) to dta: "Block", "Row" and "Col" (case sensitive), where Block may refer to a field block or an enviroment without blocks Therefore, make sure that blocks from different environments are named differently. Row and Col provide the coordinates for the identification of neighbor plots.
it	Integer. Total numeric of MCMC iterations used to fit the model.
bi	Integer. Burn-in of MCMC iterations, i.e., number of iteration to be discarted prior to model convergence.
th	Integer. Thinning parameter: saves only 1 interation every th. Thinning is used to reduce the auto-correlation between Markov chains.
model	Prediction model: The options are: BRR, BayesA, GBLUP, RKHS and RF.
...	Pass arguments to the function that builds the spatial splines NNsrc: rho and dist. By default, rho=1 and dist=3. To check how it looks like in the field, type NNsrc(rho=1,dist=3). If model="RF", it also passes the setting of random forest: number of trees (NTR=50), sample size (SSZ=80), max nodes (MNO=10), node size (NSZ=4), number of variables (MTR=50), importance sampling (IMP=TRUE) and replacement (RPL=TRUE).

Details

The general model is $y = Xb + Zu + f(x) + e$, where y is the response variable, Xb refers to the fixed effects, Zu regards the genetic effect, $f(x)$ represents the field variation, and e is the vector of residuals. In this model u is a link function that represents the genetic component, which depends on the model specified.

For whole-genome regression models (BRR or BayesA), $u = Ma$, where M is the matrix of genotypes. For kernel models (RKHS and GBLUP), $u = N(0, K\sigma 2a)$, where K is either a Gaussian kernel (RKHS) or a linear kernel (GBLUP). For the non-parametric model (RF), u is updated every round of MCMC and prediction on unobseved genotyped are performed in every round of MCMC and average out in the end. To avoid over-representation of genotypes, u is not weighted according to the number of observations of each genotype.

Unobserved genotypes not provided in dta but provided in gen are predicted in the output of the function. Genotypes without genotypic information are transfered to the fixed effect (eg. checks). Missing loci are imputed with the expectation. If dta is not provided, the function will work as a regular genomic prediction model, so the length of y must match the number of rows of gen.

In whole-genome regression models, the regularization of the genetic term is either based on chosen prior (t, Gaussian), Gaussian (from ridge regression) and t (from BayesA). Kernel models (GBLUP and RKHS) are regularized as Gaussian process, which is similar to a ridge regression of Eigenvectors where the regularization of Eigenpairs also relies on the Eigenvalues. Random forest is, at some extend, regularized by multiple parameters including sample size (SSZ), numer of trees (NTR), number of variables (MTR), etc.

If there is a large number of trials and users acknowledge the necessity of sparse matrices, we recommend installing the Matrix package and run the following code that enables sparsity:

```
source(system.file("add","sparseGMM.R",package="NAM"))
```

Value

The function gmm returns a list containing the fitted values (hat), observed values (obs), intercept (mu, incidence matrix of genotypes (Z) and the vector of breeding values (EBV). If fixed effects are provided, it also returns the design matrices and coefficients of fixed effects (X,b). If the model was kernel or regression, output will include the random effect coefficients (g), variance components of markers (Vg) and residuals (Ve). Kernel models regress the Eigenvectors of the kernel, weighted by the Eigenvalue. The coefficient (cxx) used in the BRR model to convert marker variance Vb into genetic variance Va. If spatial information is provided, the output includes the fitted spatial term (sp).

Author(s)

Alencar Xavier

Examples

```
# For a demo with wulti-environmental trial type:
# demo(fittingMET)

# Checking heritability
data(tpod)
fit = gmm(y,gen,model = 'BRR')
fit$Vg*fit$cxx / (fit$Vg*fit$cxx+fit$Ve)
```

Optimizations *Internal functions*

Description

Internal function under optimization, complimentary statistics, and loops written in C++ to speed up $gwas$, $gibbs$ and wgr.

Details

Some of the functions available for users include:

01) Import_data(file,type=c('GBS','HapMap','VCF')): This function can be used to import genotypic data in the NAM format, providing a list with a genotypic matrix gen coded as 012 and a vector chr with count of markers per chromosome. Currently, it helps users to import three types of files: GBS text, HapMap and VCF.

02) markov(gen,chr): Imputation method based forwards Markov model for SNP data coded as 012. We recommend users to remove non-segregating markers before using this function.

03) LD(gen): Computes the linkage disequilibrium in terms of r2 for SNP data coded as 012. Missing data is not allowed.

04) PedMat(ped): Builds a kinship from a pedigree. Input format is provided with PedMat().

05) `PedMat2(ped,gen=NULL,IgnoreInbr=FALSE,PureLines=FALSE)`: Builds a kinship from a genomic data and pedigree. Useful when not all individuals are genotyped. Row names of `gen` must indicate the genotype id.

06) `Gdist(gen, method = 1)`: Computes genetic distance among individuals. Five methods are available: 1) Nei distance; 2) Edwards distance; 3) Reynolds distance; 4) Rogers distance; 5) Provesti's distance. 6) Modified Rogers distance

07) `covar(sp=NULL,rho=3.5,type=1,dist=2.5)`: Builds a spatial kernel from field plot information. Input format is provided with `covar()`. Parameter `rho` detemines the decay of relationship among neighbor plots. `type` defines if the kernel is exponential (1), Gaussian (2) or some intermediate. `dist` informs the distance ratio between range neighbors and row neighbors.

08) `eigX(gen,fam)`: Computes the input of the argument `EIG` of the function `gwas2`.

09) `G2A_Kernels(gen)`: Computes a list of orthogonal kernels containing additive, dominant and first-order epistatic effects, in accordance to the G2A model from ZB Zeng et al. 2005. These kernels can be used for description of genetic architecture through variance components, for that we recommend packages `varComp` and `BGLR`.

10) `NNsrc(sp=NULL,rho=1,dist=3)`: Using the same field data input required by the function `covar`, this function provides a list of nearest neighbor plots for each entry.

11) `NNcov(NN,y)`: This function utilizes the output of `NNsrc` to generate a numeric vector, averageing the observed values of `y`. This function is useful to generate field covariates to control micro-environmental variance without krigging.

11) `emXX(y,gen,...)`: Fits whole-genome regressions using the expectation-maximization algorithm as opposed to MCMC. Currently avaible methods include BayesA (`emBA`), BayesB (`emBB`), BayesC (`emBC`), BLASSO (`emBL`), BLASSO2 (`emDE`), Elastic-Net (`emEN`), maximum likelihood (`emML`), and Ridge regression (`emRR`). A cross-validation option is also available (`emCV`). Vide package bWGR for the documentation of these functions.

12) `CNT(X)`: Centralizes parameters from matrix `X`.

13) `IMP(X)`: Imputes missing points from matrix `X` with the average value of the column.

14) `GAU(X)`: Creates a Gaussian kernel from matrix `X`.

15) `GRM(X, Code012=FALSE)`: Creates genomic relationship matrix as linear kernel from matrix `X`. If genotypes are coded as 012 and Code012=TRUE, the kinship is the same as proposed by VanRaden (2008), otherwise the outcome is an additive G2A kernel.

16) `MSX(X)`: Computes the cross-product of each column of `X` and the sum of variances of each column of `X`.

17) `NOR(y,X,cxx,xx,maxit=50,tol=10e-6)`: Solves a ridge regression using GSRU, where `y` corresponds to the response variable, `X` is the set of parameters, `cxx` and `xx` are the output from the MSX function, `maxit` and `tol` are the convergence criteria.

18) `SPC(y,blk,row,col,rN=3,cN=1)`: Computes a spatial covariate, similar to what could be obtained using NNsrc and NNcov but in a single step. It often is faster than NNsrc/NNcov.

19) `SPM(blk,row,col,rN=3,cN=1)`: Computes a spatial matrix that capture nearest neighbots, to be used as design matrix of random effects. The least-square solution gives the same as SPC.

20) `BRR2(y,X1,X2,it=1500,bi=500,df=5,R2=0.5)`: A simple C++ implementation of a Bayesian Ridge Regression that accomodates two random effects.

21) `press(y,K,MaxIt=10)`:Solves a PRESS-regularized GBLUP. You can provide K as a matrix or as the output of the functin eigen. MaxIt the maximum number of iterations to for updating missing values (if any) if H*y does not converge.

22) `emGWA(y,gen)`: A vanilla algorithm written in C++ for GWAS (very simple, but very efficient). It fits a snpBLUP via EM-REML based GSRU, then run an additional round checkinkg the likelihood of treating each marker as fixed effect instead of random, thus avoiding double-fitting. It returns the marker p-values, snpBLUP marker effects for genomic prediction, LS marker effects from the GWAS, variance components, heritability, and GEBVs (fitted values).

23) `BCpi(y,X,it=3000,bi=500,df=5,R2=0.5)`: A vanilla implementation in C++ of BayesCpi for GWAS or GWP. It returns the marker p-values (as the minus log probability of marker excluded), marker effects for genomic prediction, probability of marker included, variance components, heritability, and GEBVs (fitted values).

Author(s)

Alencar Xavier and Tiago Pimenta

Examples

```
# Forward gen imputation
data(tpod)
fast.impute = markov(gen,chr)

# Wright's A matrix
PedMat()

# Pairwise LD
ld = LD(gen[,1:10])
heatmap(ld)

# Spatial correlation (kernel-based)
covar()

# Spatial correlation (NN-based)
NNsrc()

# Genetic distance
round(Gdist(gen[1:10,],method=1),2)

# PCs of a NAM kinship
eG = eigX(gen,fam)
plot(eG[[2]],col=fam)

# Polygenic kinship matrices
Ks = G2A_Kernels(gen)
ls(Ks)

# Genomic regression fitted via EM
h = emBA(y,gen)
plot(h$b,pch=20)
```

```
# GBLUP and RRBLUP
g = GRM(gen)
eg = eigen(g)
gblup = emML(y=y, gen=eg$vectors,D=eg$values)
rrblup = emML(y=y, gen=gen)
plot(gblup$hat,rrblup$hat,xlab = 'gblup',ylab='rrblup')

# Vanilla GWAS
gwa = emGWA(y,gen)
plot(gwa$PVAL,pch=20)
```

SNP H2 *SNP heritability*

Description

Calculates the ability of markers to carry a gene (Foneris et al. 2015). The index is also an indicator of Mendelian segregation.

Usage

```
snpH2(gen,K=NULL)
```

Arguments

gen	Numeric matrix containing the genotypic data. A matrix with n rows of observations and (m) columns of molecular markers.
K	Optional. Numeric matrix containing the genetic relationship matrix. A square and symmetric matrix $(n$x$n)$ with the same observations as the matrix gen, also in the same order. If not provided, the kinship matrix is estimated from the genotypic matrix.

Value

Numeric vector containing the heritability of each markers. Foneris et al. (2015) recommends to avoid using markers with index lower than 0.98.

Author(s)

Alencar Xavier

References

Forneris, N. S., Legarra, A., Vitezica, Z. G., Tsuruta, S., Aguilar, I., Misztal, I., & Cantet, R. J. (2015). Quality Control of Genotypes Using Heritability Estimates of Gene Content at the Marker. Genetics 199(3):675-681.

Examples

```
data(tpod)
Heritability=snpH2(gen)
plot(Heritability,chr=chr)
```

SNP QC *SNP Quality Control*

Description

Functions for quality control. 'snpQC' may be used to count/remove neighbor repeated SNPs, markers with MAF lower than a given threshold, and imputations. 'cleanREP' identifies and merge duplicate genotypes. The 'reference' function changes the reference genotype. For NAM populations, this function must be used when genotypes are coded according to the reference genome instead of the standard parent.

Usage

```
snpQC(gen,psy=1,MAF=0.05,misThr=0.8,remove=TRUE,impute=FALSE)
cleanREP(y,gen,fam=NULL,thr=0.95)
reference(gen,ref=NULL)
```

Arguments

gen	Numeric matrix containing the genotypic data. A matrix with n rows of observations and (m) columns of molecular markers. SNPs must be coded as `0, 1, 2`, for founder homozigous, heterozigous and reference homozigous. `NA` is allowed.
psy	Tolerance parameter for markers in Perfect SYymmetry (psy). This QC remove identical markers (aka. full LD) that carry the same information. Default is 1, which removes only SNPs 100% equal to its following neighbor.
MAF	Minor Allele Frequency. Default is 0.05. Useful to inform or remove markers below the MAF threshold. Markers with standard deviation below the MAF threshold will be also removed.
misThr	Missing value threshold. Default is 0.8, removing markers with more than 80 percent missing values.
remove	Logical. Remove SNPs due to PSY or MAF.
impute	If TRUE, impute missing values using Random Forest adapted from the package missForest (Stekhoven and Buhlmann 2012) as suggested by Rutkoski et al (2013).
y	Numeric vector (n) or numeric matrix (n x t) of observations describing the trait to be analyzed. `NA` is allowed.
fam	Numeric vector of length (n) indicating which subpopulations ($i.e.$ family) each observation comes from. Default assumes that all observations are from the same populations.

`thr` Threshold above which genotypes are considered identical. Default is 0.95, merging genotypes >95 percent identical.

`ref` Numeric vector of length n with elements coded as `0, 1, 2`, it represents the genotypic information of a new reference genotype. Default assumes that more frequent allele represents the reference genome.

Value

snpQC - Returns the genomic matrix without missing values, redundancy or low MAF markers.

cleanREP - List containing the inputs without replicates. Groups of replicates are replaced by a single observation with the phenotypic expected value. The algorithm keeps the genotypic information of the first individual (genotypic matrix order).

reference - Returns a recoded gen matrix

Author(s)

Alencar Xavier, Katy Rainey and William Muir

References

Rutkoski, J. E., Poland, J., Jannink, J. L., & Sorrells, M. E. (2013). Imputation of unordered markers and the impact on genomic selection accuracy. G3: Genes| Genomes| Genetics, 3(3), 427-439.

Stekhoven, D. J. and Buhlmann, P. 2012. MissForest - nonparametric missing value imputation for mixed-type data. Bioinformatics, 28(1), 112-118.

Examples

```
data(tpod)
gen=reference(gen)
gen=snpQC(gen=gen,psy=1,MAF=0.05,remove=TRUE,impute=FALSE)
test=cleanREP(y,gen)
```

Index