Package 'MixABEL'

March 18, 2011

Type Package

Title mixed models for genetic association analysis

Version 0.1-1

Date 2011-02-23

Author Yurii Aulchenko, William Astle, Erik Roos, Marcel Kempenaar

Maintainer Yurii Aulchenko <i.aoultchenko@erasmusmc.nl>

Depends R (>= 2.4.0), MASS, mvtnorm, GenABEL (>= 1.5-8), DatABEL (>= 0.1-5)

Suggests

Description a package for fast mixed models for association analysis in samples of related individuals

License GPL (>= 2)

R topics documented:

`MixABEL-package` *MixABEL package...*

Description

MixABEL package

Details

Package for fast (genome-wide) association analysis using mixed models. Main functions: FastMixedModel, GWFGLS

Author(s)

Yurii Aulchenko, William Astle, Erik Roos, Marcel Kempenaar

`FastMixedModel` *fast mixed models...*

Description

fast mixed models

Usage

```
FastMixedModel(Response, Explan, Kin, Covariates, nu_naught=0,
    gamma_naught=0)
```

Arguments

`Response`	is an n dimensional vector of Responses
`Explan`	is an n*p matrix of Explanatory variables, each to be tested marginally (SNPS)
`Kin`	is the n*n kinship matrix
`Covariates`	is an n*k matrix of Covariates
`nu_naught`	(and gamma_naught) are hyperparameters which control the heaviness of the tails of the test distribution (recommend leave them unchanged).
`gamma_naught`	(and nu_naught) are hyperparameters which control the heaviness of the tails of the test distribution (recommend leave them unchanged).

Details

fast mixed models – BETA VERSION If compiled against OMP this library can exploit multi-core parallelism Does not cope with missing data at present

Value

a list with values ...

Author(s)

William Astle `<fio@where>`

References

reference to fill in

See Also

mmscore

Examples

```
require(mvtnorm)
data(ge03d2.clean)
df <- ge03d2.clean[1:250,autosomal(ge03d2.clean)]
NSNPS <- nsnps(df)
modh2 <- 0.8
gkin <- ibs(df[,autosomal(df)],w="freq")

ngkin <- gkin
ngkin[upper.tri(ngkin)] <- t(ngkin)[upper.tri(ngkin)]
ngkin[1:5,1:5]
mysig <- (modh2*2*ngkin+(1.-modh2)*diag(dim(ngkin)[1]))
mysig[1:5,1:5]
mytra <- as.vector(rmvnorm(1,sigma=mysig)) + phdata(df)$sex*0.05 + phdata(df)$age*0.002
mytra[1:10]
df@phdata$mytra <- mytra
df@phdata[1:5,]

time0.h2 <- proc.time()
h2 <- polygenic(mytra~sex+age,data=df,kin=gkin)
time.h2 <- proc.time() - time0.h2

time0.mms <- proc.time()
mms <- mmscore(h2,data=df)
time.mms <- proc.time() - time0.mms

time0.grs <- proc.time()
grs <- qtscore(h2$pgres,data=df)
time.grs <- proc.time() - time0.grs

res <- mytra
summary(res)
expl <- as.numeric(df[,1:NSNPS])
summary(res)
covariates <- matrix(c(phdata(df)$sex,phdata(df)$age),ncol=2)
summary(covariates)

time0.fmm <- proc.time()
fmm <- FastMixedModel(Response=res,
Explan=expl,
Kin = gkin,
Cov=covariates)
time.fmm <- proc.time() - time0.fmm

time.h2
time.h2+time.grs
time.h2+time.mms
time.fmm

h2$h2an
#mms$effB
#mms$chi2.1df
fmm$null.herit

cor(mms[,"chi2.1df"],fmm$chi.sq)^2
plot(mms[,"chi2.1df"],fmm$chi.sq)
```

FGLS *Feasible GLS...*

Description

Feasible GLS

Usage

```
FGLS(Y, X, W, test="wald", whichtest=c(FALSE, rep(TRUE, dim(X)[2] -
    1)))
```

Arguments

Y	dependent variable
X	design matrix (including intercept, if necessary)
test	test to be applied, one of 'wald', 'score' or 'robust'
whichtest	which independent variables to be tested (set to 'TRUE')
W	for GLS, inverse variance-covariance matrix, as such returned by GenABEL's polygenic(...)$InvSigma, or NULL for LS

Details

Feasible Generalised Least Squares

Value

List with elements 'beta' – estimates fo the regression coefficients; 'V' – variance covariance matrix for parameters estimates; 'T2' – test statistics (distributed as Chi-squared under the null) for the testing of whichtest parameters; 'df' – the number of degrees of freedom of the T2 test; 'tested' – which parameters were tested with T2; 'meanColX' – mean value of variable in columns of X; 'n' – length of Y (== height of X)

Author(s)

Yurii Aulchenko

GWFGLS *Genome-wide FGLS...*

Description

Genome-wide FGLS

Usage

```
GWFGLS(formula, data, subset, weights, na.action, contrasts, offset, W,
    inverse=TRUE, na.SNP="impute", mincall=0.95, residuals=FALSE,
    test="wald", model.SNP="additive", genodata, gtcoding="typed",
    verbosity=1, varcov=FALSE, include.means=TRUE, singular="ignore",
    with.lm=FALSE, old=FALSE)
```

Arguments

formula	analysis formula; should contain special 'SNP' term
data	phenotypic data frame, or 'gwaa.data-class' object
subset	subset of data (individuals)
weights	RESERVED FOR FURTURE USE
na.action	RESERVED FOR FURTURE USE
contrasts	RESERVED FOR FURTURE USE
offset	RESERVED FOR FURTURE USE
W	for GLS, (inverse of) variance-covariance matrix, as such returned by GenABEL's polygenic(...)\$InvSigma, or NULL for LS
inverse	wheather W is already inverted
na.SNP	how to deal with missing SNP data; 'impute' – substitute with mean, 'drop' – drop rows with missing genotypes
mincall	minimall call rate for a SNP (if less, the SNP is dropped)
residuals	use residuals for analysis? (faster, but less precise)
test	test to be applied, one of 'wald', 'score' or 'robust'
model.SNP	SNP model to apply, one of c("additive","dominantB","recessiveB","overdominant","genotypic")
genodata	genotypic data; can be missing if data is of 'gwaa.data-class'. Otherwise can be regular matrix or 'databel' matrix
gtcoding	one of c("typed","dose","probability") 'typed' – coded with NA, 0, 1, 2
verbosity	what to report; 0 – only test stats, 1 – test stats and estimates concerning 'whichtest' parameters, 2 – test stats and estimates of all parameters
varcov	wheather var-cov matrix for estimated parameters is to be reported
include.means	weather mean values of predictors are to be reported
singular	what to do with linear dependencies in X (now only 'ignore' implemented)
with.lm	whether LM should be run along (only test purposes; always falls back to 'old' R-only implementation)
old	if TRUE, old R-only code implementation is running (testing purposes, slow)

Details

Genome-wide Feasible GLS

Author(s)

Yurii Aulchenko

Examples

```
library(MASS)
library(mvtnorm)
library(GenABEL)
data(ge03d2.clean)
NIDS <- 100
df <- ge03d2.clean[1:NIDS,autosomal(ge03d2.clean)]
s <- summary(df@gtdata)
```

```
maf <- pmin(s$Q.2,1-s$Q.2)
df <- df[,(maf>0.05)]
gkin <- ibs(df[,autosomal(df)],w="freq")

modelh2 <- 0.8
covars <- c("sex","age")
s2 <- 12^2
betas_cov <- c(170,12,0.01)

X <- as.matrix(cbind(rep(1,NIDS),phdata(ge03d2.clean)[1:NIDS,covars]))
grel <- gkin
grel[upper.tri(grel)] <- t(grel)[upper.tri(grel)]
grel <- 2*grel
Y <- as.vector(rmvnorm(1,mean=(X %*% betas_cov),sigma=s2*(modelh2*grel+diag(dim(grel)[1])
length(Y)
Y[2] <- NA
df@phdata$Y <- Y

mdl <- Y~age+sex
h2 <- polygenic(mdl,data=df,kin=gkin)
h2$h2an
mdl <- Y~age+sex+SNP

gtOld <- df@gtdata[,1:10]
gtReal <- as.double(gtOld)
gtNew <- as(gtReal,"databel")
aIR <- GWFGLS(mdl,data=phdata(df),genodata = gtReal,verbosity=0) #,model="genotypic")
aIO <- GWFGLS(mdl,data=phdata(df),genodata = gtOld, verbosity=0) #,model="genotypic")
aIN <- GWFGLS(mdl,data=phdata(df),genodata = gtNew, verbosity=0) #,model="genotypic")
aWR <- GWFGLS(mdl,data=phdata(df),W=h2$InvSigma,genodata = gtReal,verbosity=0) #,model="g
aWO <- GWFGLS(mdl,data=phdata(df),W=h2$InvSigma,genodata = gtOld, verbosity=0) #,model="g
aWN <- GWFGLS(mdl,data=phdata(df),W=h2$InvSigma,genodata = gtNew, verbosity=0) #,model="g
aIN
aWN
complex_model <- GWFGLS(Y~age+sex*SNP,data=phdata(df),W=h2$InvSigma,
genodata = gtReal,verbosity=2,varcov=TRUE,include.means=TRUE,model="genotypic")
complex_model
```

`summaryFGLS` *summary for FGLS...*

Description

summary for FGLS

Usage

```
summaryFGLS(x, verbosity=1, varcov=FALSE, include.means=FALSE)
```

Arguments

`x`	object returned by FGLS function
`verbosity`	what to report; 0 – only test stats, 1 – test stats and estimates concerning 'whicht-est' parameters, 2 – test stats and estimates of all parameters

`varcov` wheather var-cov matrix for estimated parameters is to be reported

`include.means` weather mean values of predictors are to be reported

Details

summary for Feasible GLS

Value

one-line summary of output from FGLS function

Author(s)

Yurii Aulchenko

Index