Package ‘AFfR’

October 9, 2020

Type Package

Title Added Functions for R.ME

Version 0.0.2

URL http://github.com/yzhlinscau/AFfR

BugReports http://github.com/yzhlinscau/AFfR/issues

Depends R (¿= 4.0.0), crayon

Suggests agricolae,amap, desplot, dplyr, genetics, GeneticsPed, ggplot2, lme4, Matrix, msm, plyr, reshape2,quantmod, R.rsp,scales,sommer, zoo

Enhances asreml, breedR

VignetteBuilder R.rsp

Date 2020-10-07

Author Yuanzhen Lin [cre, aut]

Maintainer Yuanzhen Lin <yzhlinscau@163.com>

Description This package adds some functions added to asreml (V4 & V3) and breedR packages, functions including: asreml.batch()to carry out batch analysis in ASReml-R, pin() or pin4() to count heritability or correlation with se and output significant level for correlation value, model.comp() to compare models with different randomed factors for asreml, plot1() and met.plot() to plot spatial and MET data, met.corr() to calculate var/cov/corr matrix for MET analysis.

License GPL (¿= 2)

LazyData yes

NeedsCompilation no

Repository CRAN

RoxygenNote 7.1.1

R topics documented:

`AF.asreml` *Summary of added functions for asreml*

Description

Summary of added functions for asreml

Details

This package would supply some functions for asreml. Details as following:

Function	**Description**
`asreml.batch`	ASReml-R batch analysis.
`AGH.inv`	generate H-inverse matrix, etc.
`coef2`	Extract model effects with standard errors.
`pin`	Count error for h2 and corr.
`pin.batch`	Run pin batch for one object.
`pin2.batch`	Run pin batch for more than two objects.
`repin`	run repin for batch result.
`update`	rerun new batch from old batch result.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

`AF.base` *Summary of some R basic functions*

Description

Summary of some R basic functions

Details

This package would supply some functions for R. Details as following:

Function	**Description**
`datatable2`	Create an HTML table widget.
`mc.run`	multi-core run in R.
`heatmap1`	Create a Heatmap.
`plot1`	Plotting test trait's norm or multi-comparison.
`spd.plot`	Plot spatial data or Variogram.
`Var`	Output Variance components for R packages.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

AF.breedR *Summary of added functions for breedR*

Description

Summary of added functions for breedR

Details

This package would supply some functions for breedR. Details as following:

Function	Description
pin	run pin for breedR.
pin.batch	run batch-pin for breedR.
breedR.batch	run batch for breedR.
repin	run repin for breedR batch.
update	Orun update for breedR batch.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

AF.Echidna *Summary of added functions for Echidna*

Description

Summary of added functions for Echidna

Usage

```
esRT(trace = FALSE)
```

Details

This package would supply some functions for Echidna. Details as following:

Function	Description
wald	output wald results.
Var	output variance components.
summary	output summary results.
IC	output AIC and BIC values.
pin	output pin results.
predict	output predict results.
plot	output model diagnose results.
coef	output fixed and random effects.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
## Not run:

 library(AFfR)

 ##  Echidna
 path='D:\\Echidna\\Jobs\\METb11'
 setwd(path)

 rm(res)
 dir()

 #  mainly works for '_e.R'
 res<-esRT(trace=T)

 names(res)
 class(res)

 # check .esy exist or not
 plot(res)

 # wald result
 wald(res)
 #wald(res,c(-3,-5,-6))

 # variance components
 Var(res)

 # summary result
 summary(res)

 # AIC,BIC result
 IC(res)

 # fixed and random effects
 coef(res)$fixed
 coef(res)$random

 # predict results if using predict functions
 mm<-predict(res)
 mm$pred

 # pin results if using vpredict function
 pin(res)

 # model converge stage
 trace(res)
 res$Converge


## End(Not run)
```

AF.sommer *Summary of added functions for sommer*

Description

Summary of added functions for sommer

Details

This package would supply some functions for sommer. Details as following:

Function	**Description**
predict2	run predict for sommer.
pin	run pin for sommer.
pin.batch	run batch-pin for sommer.
sommer.batch	run batch for sommer.
repin	run repin for sommer batch.
update	Orun update for sommer batch.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

AGH.inv *Generate H-inverse matrix for SS-GBLUP.*

Description

AGH.inv This function calculate genomic relationship matrix(G), full additative matrix(A) and blended relationship matrix(H) from genotyped marker, genotyped pedigree and ungenotyped pedigree.

Usage

```
AGH.inv(option=1, ugped,gped,gmarker)
```

Arguments

option	option (1˜5) for different G matrixs.
ugped	ungenotyped pedigree, or total pedigree.
gped	genotyped pedigree.
gmarker	genotyped marker,column 1 should be sample ID.

Details

This function would return a list containing 3 elements. The types of option (1~5) as following:

option	Description
1	observed allele frequencies (GOF, VanRaden, 2008).
2	weighted markers by recipricals of expected variance (GD, Forni et al., 2011).
3	allele frequencies fixed at 0.5 (G05, Forni et al., 2011).
4	allele frequencies fixed at mean for each locus (GMF, Forni et al., 2011).
5	regression of MM' on A sort (Greg, VanRaden, 2008).

Value

Ainv inverse of full additative matrix(A).

Ginv inverse of genomic relationship matrix(G).

Hinv inverse of blended relationship matrix(H).

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
library(AFfR)

data("ugped")
data("gped")
data("gmarker")

# get A-matrix, G-matrix and H-matrix
AGH1<-AGH.inv(option=1,ugped,gped,gmarker)


data(MET)
MET$yield<-0.01*MET$yield
levels(MET$Genotype)<-gped$ID
MET1<-filterD1(MET, Loc %in% c(3))

## for ASReml-R V3.0
library(asreml)

# base model
sm1.asr<-asreml(yield~Rep, random=~ Genotype+units,
                rcov=~ ar1(Col):ar1(Row),
                data=MET1, maxiter=50)

Var(sm1.asr)

# A-BLUP
Ainv <- AGH1$Ainv
```

```
sm2.asr<-update(sm1.asr, random=~ ped(Genotype)+units,
                ginverse=list(Genotype=Ainv))

Var(sm2.asr)

# G-BLUP
Ginv <- AGH1$Ginv
sm3.asr<-update(sm1.asr, random=~ ped(Genotype)+units,
                ginverse=list(Genotype=Ginv))

Var(sm3.asr)

# H-BLUP
Hinv <- AGH1$Hinv
sm4.asr<-update(sm1.asr, random=~ ped(Genotype)+units,
                ginverse=list(Genotype=Hinv))

Var(sm4.asr)


## for ASReml-R V4
library(asreml)


sm1.asr<-asreml(yield~Rep, random=~ Genotype+units,
                residual=~ ar1(Col):ar1(Row),
                data=MET1, maxiter=50)

Var(sm1.asr)

# A-BLUP
Ainv <- AGH1$Ainv
sm2.asr<-update(sm1.asr,
             random=~ vm(Genotype,Ainv)+units)

Var(sm2.asr)

# G-BLUP
Ginv <- AGH1$Ginv
sm3.asr<-update(sm1.asr,
             random=~ vm(Genotype,Ginv)+units)

Var(sm3.asr)

# H-BLUP
Hinv <- AGH1$Hinv
sm4.asr<-update(sm1.asr,
             random=~ vm(Genotype,Hinv)+units)

Var(sm4.asr)

########## if any other genotyped id without ped
########## we can put their parent code to 0 or NA
########## to make pedigree, then use H-matrix.
## gmarker2 without pedigree
#
## make their pedigree
```

```
# gid2<-gmarker2[,1]
# gped2<-data.frame(ID=gid2,Female=0,Male=0)
#
## combine genotyped id's pedigree
# gped1<-rbind(gped,gped2)
#
## combine all genotyped marker data
# gmarker1<-rbind(gmarker,gmarker2)
#
# AGH1a<-AGH.inv(option=1,tped1,gped1,gmarker1)
#
```

`asreml.batch2` *ASReml-R batch analysis*

Description

`asreml.batch2` This function carries out batch analysis for multi-trait with the same model and also output heritability etc. in ASReml-R package. Almost all models are also avaible in the `asreml.batch2`. User also can store all results for each run when carry out batch analysis with parameter ’all.result=TRUE’, then the stored object, such as ’run1’, can be re-used for further counting new parameters by `repin` or running new batch by `update2`.

Usage

```
asreml.batch2(data, factorN, traitN,
                      FMod=NULL, RMod=NULL, EMod=NULL,
                      mulT=NULL, mulN=NULL, mulRM=NULL,
                      corM=NULL, corMout=FALSE,
                      order=FALSE, plot.matrix=FALSE, all.result=FALSE,
                      rgFormula=NULL, mulp=NULL, maxit=NULL, digit=3,
                      family=NULL, ped=NULL, pedinv=NULL, ginverse=NULL)
```

Arguments

`data` aim dataset

`factorN` A vector with sites of all factors.

`traitN` A vector with sites of all traits.

`FMod` Fixed mode.

`RMod` Randomed variance structure (G structure).

`EMod` Error variance structure (R) for multi-trait model.

`mulT` Value ”T” or ”TRUE” for multi-trait model, ”F”(default).

`mulN` Number of trait for one model in multi-trait analysis,2(default).

`mulRM` Logical Value ”T” or ”TRUE” to count corr/error matrix, only works for bitrait, ”F”(default).

`corM` Logical Value ”T” or ”TRUE” for corr model, ”F”(default).

`corMout` Logical Value ”T” or ”TRUE” to output corr matrix, ”F”(default).

`order` Logical Value ”T” or ”TRUE” to reorder corr matrix, ”T”(default).

`plot.matrix` Logical Value ”T” or ”TRUE” with corMout (also ’T’)to output corr matrix for plotting, ”F”(default).

`all.result` Logical Value ”T” or ”TRUE” to return each asreml run’s all results,”F”(default).

`rgFormula` formula for corr, etc.

`mulp` a list of formula for h2 or corr,c(h2˜4*V1/(V1+V2)).

`maxit` Maximum number of iterations.

`digit` index to control decimal number.

`family` A list of functions and expressions for defining the link and variance functions.

`ped` Value ”T” or ”TRUE” for animal model with pedigree, ”F”(default).

`pedinv` A G-inverse matrix for pedigree from ainverse() for V4 or V3.

`ginverse` A named list with each component identifying a G-inverse matrix, only needed for V3.

Details

ASReml-R batch analysis

Value

the result is returned directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

See Also

update repin

Examples

```
## Not run:

library(AFfR)
library(RSTAT2D)

##### example I for dataset without pedigree in ASReml-R
library(asreml) # for V3.0, or V4.1
#library(asreml4) # for V4.0

data(PrSpa,package='AFfR')
df<-PrSpa

# exmaple 1.1 single trait model
df1<-subset(df,Spacing==3)

asreml.batch2(data=df1,factorN=1:5,traitN=c(9:13),
          FMod=y~1+Rep+Plot,
          RMod=~Fam,
          #family=asreml.gaussian(), # for V3
          #family=asr_gaussian(),    # for V4
          mulp=c(h2 ~ 4 * V1/(V1+V2)))


# exmaple 1.2 us model
asreml.batch2(data=df1,factorN=1:5,traitN=c(10:13),
          FMod=cbind(y1,y2)~trait+trait:Rep,
          RMod=~us(trait):Fam,
          EMod=~units:us(trait),
          #family=c(asreml.gaussian(),asreml.gaussian()), # for V3
          #family=list(asr_gaussian(),asr_gaussian()),    # for V4
          mulT=TRUE,mulN=2,mulRM=TRUE,corMout=TRUE,
          mulp=c(gcorr ~ V2/sqrt(V1*V3),
                 h2.A ~ 4*V1/(V1+V5),
                 h2.B ~ 4*V3/sqrt(V3+V7)))


# exmaple 1.3 corr model
asreml.batch2(data=df1,factorN=1:5,traitN=c(10:13),
             FMod=cbind(y1,y2,y3)~trait+trait:Rep,
             RMod=~corgh(trait):Fam,
             EMod=~units:us(trait),
             maxit=30,mulT=TRUE,mulN=3,corM=TRUE)


##### example II for dataset with pedigree in ASReml-R
data(dfm2,package='RSTAT2D')

ped<-dfm2[,1:3]
#ainv<-asreml.Ainverse(ped)$ginv # for V3
ainv<-ainverse(ped)  # for V4 or V3

dfm2a<-subset(dfm2,Spacing==3)

# example 2.1 single trait model
asreml.batch2(data=dfm2a,factorN=1:6,traitN=c(7:14),
          FMod=y~1+Rep,
          RMod=~ped(TreeID),      # for V3
          #RMod=~vm(TreeID,ainv), # for V4
          ped=TRUE,pedinv=ainv,
          ginverse=list(TreeID=ainv), # only need for V3
          mulp=c(h2 ~ V1/(V1+V2)))

# exmaple 2.2 us model
asreml.batch2(data=dfm2a,factorN=1:6,traitN=c(10:14),
             FMod=cbind(y1,y2)~trait+trait:Rep,
             RMod=~us(trait):ped(TreeID), # for V3
             #RMod=~us(trait):vm(TreeID,ainv),# for V4
             EMod=~units:us(trait),maxit=40,
             mulT=TRUE,mulN=2,mulRM=TRUE,
             corMout=FALSE,
             ped=TRUE,pedinv=ainv,
             ginverse=list(TreeID=ainv), # only need for V3
             mulp=c(gcorr ~ V2/sqrt(V1*V3),
                    h2.A ~ V1/(V1+V5),
                    h2.B ~ V3/(V3+V7)))

# exmaple 2.3  corr model
asreml.batch2(data=dfm2a,factorN=1:6,traitN=c(10:14),
             FMod=cbind(y1,y2)~trait+trait:Rep,
             RMod=~corgh(trait):ped(TreeID), # for V3
             #RMod=~corgh(trait):vm(TreeID,ainv), # for V4
             EMod=~units:us(trait),maxit=40,
             mulT=TRUE,mulN=2,corM=TRUE,
             ped=TRUE,pedinv=ainv,
             ginverse=list(TreeID=ainv) # only need for V3
             )

# exmaple 3.1 spatial model for single trait
data('sp',package='AFfR')

asreml.batch2(data=sp,factorN=1:8,traitN=c(9:10),
              FMod=y~1+Rep,
              RMod=~units+Mum,
              EMod=~ar1(Row):ar1(Col),
              maxit=30,
              mulp=c(h2 ~ 4 * V1/(V1+V2)))


# exmaple 3.2  spatial model for double traits
asreml.batch2(data=sp,factorN=1:8,traitN=c(9:10),
              FMod=cbind(y1,y2)~trait+trait:Rep,
              RMod=~diag(trait):units+us(trait):Mum,
              EMod=~ar1(Row):ar1(Col):diag(trait),
              maxit=30,mulT=TRUE,mulN=2,
              mulRM=FALSE,corMout=FALSE,
              mulp=c(h2.A ~ 4 * V1/(V1+V4)))


## End(Not run)
```

batch *Batch analysis for asreml,breedR and sommer*

Description

asreml.batch,breedR.batch and sommer.batch This function carries out batch analysis for multi-trait with the same model and also output heritability etc. in ASReml-R package. Almost all models are also avaible in the asreml.batch,breedR.batch and sommer.batch. User also can store all results for each run when carry out batch analysis with parameter 'all.result=TRUE', then the stored object, such as 'run1', can be re-used for further counting new parameters by repin or running new batch by update.

Usage

```
asreml.batch(data, factorN, traitN,
                      FMod=NULL, RMod=NULL, EMod=NULL,
                      mulT=NULL, mulN=NULL, mulRM=NULL,
                      corM=NULL, corMout=FALSE,
                        order=FALSE, plot.matrix=FALSE, all.result=FALSE,
                        rgFormula=NULL, mulp=NULL, maxit=NULL,digit=3,
                        family=NULL, ped=NULL, pedinv=NULL, ginverse=NULL)

breedR.batch(data,factorN, traitN,
                          FMod=NULL,RMod=NULL,
                          genetic=NULL,spatial=NULL,
                          generic = NULL,
                          weights = NULL,method = 'ai',
                          maxit=30,trace=FALSE,
                          mulT=F,mulN=NULL,
                          mulp=NULL,all.result=F)

sommer.batch(data,factorN, traitN,
                          FMod=NULL,RMod=NULL,EMod=NULL,
                          maxit=30,trace=FALSE,
                          mulT=F,mulN=NULL,
                          mulp=NULL,all.result=F)
```

Arguments

`data` aim dataset

`factorN` A vector with sites of all factors.

`traitN` A vector with sites of all traits.

`FMod` Fixed mode.

`RMod` Randomed variance structure (G structure).

`EMod` Error variance structure (R) for multi-trait model.

`mulT` Value ”T” or ”TRUE” for multi-trait model, ”F”(default).

`mulN` Number of trait for one model in multi-trait analysis,2(default).

`mulRM` Logical Value ”T” or ”TRUE” to count corr/error matrix, only works for bitrait, ”F”(default).

`corM` Logical Value ”T” or ”TRUE” for corr model, ”F”(default).

`corMout` Logical Value ”T” or ”TRUE” to output corr matrix, ”F”(default).

`order` Logical Value ”T” or ”TRUE” to reorder corr matrix, ”T”(default).

`plot.matrix` Logical Value ”T” or ”TRUE” with corMout (also ’T’)to output corr matrix for plotting, ”F”(default).

`all.result` Logical Value ”T” or ”TRUE” to return each asreml run’s all results,”F”(default).

`rgFormula` formula for corr, etc.

`mulp` a list of formula for h2 or corr,c(h2˜4*V1/(V1+V2)).

`maxit` Maximum number of iterations.

`digit` index to control decimal number.

`family` A list of functions and expressions for defining the link and variance functions.

`ped` Value ”T” or ”TRUE” for animal model with pedigree, ”F”(default).

`pedinv` A G-inverse matrix for pedigree from ainverse() for V4 or V3.

`ginverse` A named list with each component identifying a G-inverse matrix, only needed for V3.

genetic	if not NULL, a list with relevant parameters for an additive genetic effect.
spatial	if not NULL, a list with relevant parameters for a spatial random effect.
generic	if not NULL, a named list with an incidence matrix and either a covariance or a precision matrix
weights	numeric. A vector of weights for the residual variance.
method	either 'ai' or 'em' for Average-Information or Expectation-Maximization REML respectively

Details

Batch analysis for asreml,breedR and sommer

Value

the result is returned directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

See Also

update
repin

Examples

```
## Not run:

library(AFfR)
library(RSTAT2D)

##### example I for dataset without pedigree in ASReml-R
library(asreml) # for V3.0, or V4.1
#library(asreml4) # for V4.0

data(PrSpa,package='AFfR')
df<-PrSpa

# exmaple 1.1 single trait model
df1<-subset(df,Spacing==3)

asreml.batch(data=df1,factorN=1:5,traitN=c(9:13),
          FMod=y~1+Rep+Plot,
          RMod=~Fam,
          #family=asreml.gaussian(), # for V3
          #family=asr_gaussian(),    # for V4
          mulp=c(h2 ~ 4 * V1/(V1+V2)))


# exmaple 1.2 us model
```

```
asreml.batch(data=df1,factorN=1:5,traitN=c(10:13),
          FMod=cbind(y1,y2)~trait+trait:Rep,
          RMod=~us(trait):Fam,
          EMod=~units:us(trait),
          #family=c(asreml.gaussian(),asreml.gaussian()), # for V3
          #family=list(asr_gaussian(),asr_gaussian()),    # for V4
          mulT=TRUE,mulN=2,mulRM=TRUE,corMout=TRUE,
          mulp=c(gcorr ~ V2/sqrt(V1*V3),
                 h2.A ~ 4*V1/(V1+V5),
                 h2.B ~ 4*V3/sqrt(V3+V7)))


# exmaple 1.3 corr model
asreml.batch(data=df1,factorN=1:5,traitN=c(10:13),
             FMod=cbind(y1,y2,y3)~trait+trait:Rep,
             RMod=~corgh(trait):Fam,
             EMod=~units:us(trait),
             maxit=30,
             mulT=TRUE,mulN=3,corM=TRUE)


##### example II for dataset with pedigree in ASReml-R
data(dfm2,package='RSTAT2D')

ped<-dfm2[,1:3]
#ainv<-asreml.Ainverse(ped)$ginv # for V3
ainv<-ainverse(ped)  # for V4 or V3

dfm2a<-subset(dfm2,Spacing==3)

# example 2.1 single trait model
asreml.batch(data=dfm2a,factorN=1:6,traitN=c(7:14),
          FMod=y~1+Rep,
          RMod=~ped(TreeID),      # for V3
          #RMod=~vm(TreeID,ainv), # for V4
          ped=TRUE,pedinv=ainv,
          ginverse=list(TreeID=ainv), # only need for V3
          mulp=c(h2 ~ V1/(V1+V2)))

# exmaple 2.2 us model
asreml.batch(data=dfm2a,factorN=1:6,traitN=c(10:14),
             FMod=cbind(y1,y2)~trait+trait:Rep,
             RMod=~us(trait):ped(TreeID), # for V3
             #RMod=~us(trait):vm(TreeID,ainv),# for V4
             EMod=~units:us(trait),maxit=40,
             mulT=TRUE,mulN=2,mulRM=TRUE,
             corMout=FALSE,
             ped=TRUE,pedinv=ainv,
             ginverse=list(TreeID=ainv), # only need for V3
             mulp=c(gcorr ~ V2/sqrt(V1*V3),
                    h2.A ~ V1/(V1+V5),
                    h2.B ~ V3/(V3+V7)))

# exmaple 2.3  corr model
asreml.batch(data=dfm2a,factorN=1:6,traitN=c(10:14),
             FMod=cbind(y1,y2)~trait+trait:Rep,
             RMod=~corgh(trait):ped(TreeID), # for V3
             #RMod=~corgh(trait):vm(TreeID,ainv), # for V4
             EMod=~units:us(trait),maxit=40,
             mulT=TRUE,mulN=2,corM=TRUE,
             ped=TRUE,
             pedinv=ainv,
             ginverse=list(TreeID=ainv) # only need for V3
             )

# exmaple 3.1 spatial model for single trait
data('sp',package='AFfR')

asreml.batch(data=sp,factorN=1:8,traitN=c(9:10),
              FMod=y~1+Rep,
              RMod=~units+Mum,
              EMod=~ar1(Row):ar1(Col),
              maxit=30,
              mulp=c(h2 ~ 4 * V1/(V1+V2)))


# exmaple 3.2  spatial model for double traits
asreml.batch(data=sp,factorN=1:8,traitN=c(9:10),
              FMod=cbind(y1,y2)~trait+trait:Rep,
              RMod=~diag(trait):units+us(trait):Mum,
              EMod=~ar1(Row):ar1(Col):diag(trait),
              maxit=30, mulT=TRUE,mulN=2,
              mulRM=FALSE,corMout=FALSE,
              mulp=c(h2.A ~ 4 * V1/(V1+V4)) )


## End(Not run)

## Not run:
# works for breedR
run1 <- breedR.batch(data=df1,factorN=1:5,traitN=c(9:13),
                     FMod=y~1+Rep+Plot,
                     RMod=~Fam,
                     all.result=TRUE,
                     mulp=mulp1)

## End(Not run)
## Not run:

# for sommer
data(DT_cpdata,package='sommer')
DT <- DT_cpdata
GT <- GT_cpdata
A <- sommer::A.mat(GT)

library(sommer)
res1 <- sommer.batch(data=DT,factorN=c(1,9:10),traitN=c(6:7),
             FMod=y~ 1,RMod=~ vs(id,Gu=A)+Rowf + Colf,
             EMod=~units,all.result=T,
             mulp=c(h2~V1/(V1+V4)))

sommer.batch(data=DT,factorN=c(1,9:10),traitN=c(6:7),
             FMod=cbind(y1,y2)~ 1,RMod=~ vs(id,Gu=A)+
               vs(Rowf,Gtc=diag(2)) + vs(Colf,Gtc=diag(2)),
             EMod=~vs(units),mulT=T,
             mulp=c(h2.A~V1/(V1+V8),h2.B~V3/(V3+V10)) )

## End(Not run)
```

`coef2` *Extract model effects with standard errors for asreml.*

Description

Extract model fixed or randonm effects with standard errors from an `asreml` object.

Usage

```
coef2(object, pattern, ...)

## S3 method for class 'asreml'
coef2(object, pattern = NULL, ...)
```

Arguments

`object` An `asreml` object.

`pattern` A term in the model as a character string; if an interaction then separate the term names in the string by ":". A regular expression is constructed from `pattern` to extract a subset of coefficients.

`...` Additional arguments.

Value

If `pattern` is not set, then a list of length 3 with the following components:

fixed solutions to the mixed model equations for the fixed (dense) terms.

random E-BLUPs for the effects in the random model.

sparse solutions to the mixed model equations for the fixed sparse-stored terms.

where each component is a matrix with a `dimnames` attribute.

A single matrix of effects as specified by `pattern`.

Examples

```
## Not run:
data(oats)
oats.asr <- asreml(yield ~ Variety*Nitrogen, random = ~ Blocks/Wplots, data=oats)

coef2(oats.asr)
coef2(oats.asr)$random
coef2(oats.asr, pattern="Blocks:Wplots")
coef2(oats.asr, pattern=c("Intercept","Blocks","Blocks:Wplots"))


## End(Not run)
```

datatable2 *Create an HTML table widget for dataset*

Description

datatable2 This function creates an HTML widget to display rectangular data (a matrix or data frame) using the JavaScript library DataTables.

Usage

```
datatable2(data)
```

Arguments

data	a data object (either a matrix or a data frame).

Examples

```
library(AFfR)
data('dfm',package='RSTAT2D')
datatable2(dfm)
```

dfm2 *Tree spacing trial dataset*

Description

A dataset of spacing trials of Pinus radiata

Usage

```
data(dfm2)
```

Format

one dataset with 17 variables

TreeID Tree Sample code

Mum Tree mum code

Dad Tree dad code

Spacing 3 levels

Rep Replication code

Plot Plot code

dj a numeric vector

dm a numeric vector

wd a numeric vector

h1 a numeric vector

h2 a numeric vector

h3 a numeric vector

h4 a numeric vector

h5 a numeric vector

dis a numeric vector

lt a numeric vector

str a numeric vector

`fdata` *format dataset*

Description

`fdata` This function will format dataset for Varialbes to factors.

Usage

```
fdata(data, faS = NULL, FtN = FALSE, ped = FALSE)
```

Arguments

`data`	dataset.
`faS`	The column location of factor Variables.
`FtN`	Change factoric indexs to numeric indexs, default (F).
`ped`	Change factoric pedigree to numeric type, default (F).

Details

This function formats dataset Varialbes between factor and numeric type. When using for factor to numeric, `faS` should be a list of sites for characteric and numeric type factor respectively, i.e., faS=list(c(2:4),c(1,5:8)), with site 2:4 (character factors) and site c(1,5:8) (numeric factors), one of them can be 0.

Value

this returned a formated dataset.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016 AFfR website:https://github.com/yzhlinscau/AFfR

See Also

See Also as `read.file`, `read.example`

Examples

```
library(datasets)
names(mtcars)

mtcars1<-fdata(mtcars,faS=c(2,8,9))

data(sp,package='RSTAT2D')
sp1<-fdata(sp,faS=list(c(2:4),c(1,5:8)),FtN=TRUE)
str(sp1) # or head(sp1)
# list reformed levels
sp1$rf$Mum # or sp1$rf[[1]]

# only change numeric-factor to numeric index
sp1a<-fdata(sp,faS=list(0,c(1,5:8)),FtN=TRUE)
# only change characteric-factor to numeric index
sp1b<-fdata(sp,faS=list(c(2:4),0),FtN=TRUE)

sp2<-fdata(sp[,1:3],ped=TRUE)
sp3<-fdata(sp2,faS=1:3)
```

G.data *Gblup trait data*

Description

An example dataset for GBLUP, cotaining aim traits and other factors.

Usage

```
data(G.data)
```

Format

one dataset with 7 variables

ID Sample ID

Female levels with 8

Male levels with 9

Year trial year

Site trial site

t1 trait 1

t2 trait 2

`G.marker` *Gblup marker data*

Description

An example dataset for GBLUP, cotaining id and SNPs.

Usage

```
data(G.marker)
```

Format

one genotyped dataset with 3462 variables

X Sample ID

X1 locus1,genotyped as 0, 1 and 2

X2 locus2

X3 locus3

... ...

X3460 locus3460

X3461 locus3461

`G.pedigree` *Gblup pedigree data*

Description

An example dataset for GBLUP, cotaining id, female and male.

Usage

```
data(G.pedigree)
```

Format

one dataset with 3 variables

ID Sample ID

Female levels with 8

Male levels with 9

`GenomicRel` *Generate genomic relationship matrix.*

Description

`GenomicRel` This function generates 5 genomic relationship matrixs.

option	Description
1	observed allele frequencies (GOF, VanRaden, 2008).
2	weighted markers by recipricals of expected variance (GD, Forni et al., 2011).
3	allele frequencies fixed at 0.5 (G05, Forni et al., 2011).
4	allele frequencies fixed at mean for each locus (GMF, Forni et al., 2011).
5	regression of MM' on A sort (Greg, VanRaden, 2008).

Usage

```
GenomicRel(option,data,ped=NULL)
```

Arguments

`option`	option (1~5) for different G matrixs.
`data`	markers data.
`ped`	ped data.

Author(s)

Isik Fikret

References

Isik Fikret. Genetic data analysis for plant and animal breeding. 2017

Examples

```
library(AFfR)

read.example(package = "AFfR", setpath = TRUE)
Markers<-read.file(file="sim_markers.txt",sep=' ' )
ped<-read.table( "sim_pedigree.txt", sep=' ')


 GOF1=GenomicRel( 1, Markers)
 GD1=GenomicRel( 2, Markers)
 G051=GenomicRel( 3, Markers)
 GMF1=GenomicRel( 4, Markers)

 # the same result but with ped data:
 GOF2=GenomicRel( 1, Markers, ped)
 GD2=GenomicRel( 2, Markers, ped)
 G052=GenomicRel( 3, Markers, ped)
 GMF2=GenomicRel( 4, Markers, ped)
 Greg=GenomicRel( 5, Markers, ped)
```

Ginv *Generate inverse of genomic relationship matrix.*

Description

Ginv This function generates 5 inverse of genomic relationship matrixs.

Usage

```
Ginv(marker.file, ped.file, aped.rowNames, path=NULL, Goptions=1,recal)
```

Arguments

marker.file	markers data or file.
ped.file	ped data or file.
aped.rowNames	The rowNames of pedigree inverse matrix.
path	The path of all datasets for GBLUP.
recal	recalculate G matrix with A matrix, F(default).
Goption	option (1~5) for different G matrixs.

Author(s)

Isik Fikret

References

Isik Fikret. Genetic data analysis for plant and animal breeding. 2017

Examples

```
library(AFfR)


#### 1  for demo examples
data(G.data)
data(G.pedigree)
data(G.marker)

##1.1 working for asreml V3
library(asreml) # for V3

pedinv=asreml.Ainverse(G.pedigree)$ginv
gpnames=attributes(pedinv)$rowNames

Ginv1=Ginv(marker.file=G.marker,ped.file=G.pedigree,aped.rowNames=gpnames,Goptions=1)

gblup<-asreml(t1~1+Site,random=~ giv(ID),
              ginverse=list(ID=Ginv1),data=G.data)

summary(gblup)$varcomp
c<-coef(gblup)$random
```

```
##1.2 working for asreml V4
library(asreml) # for V4

pedinv=ainverse(G.pedigree)
gpnames=attributes(pedinv)$rowNames

Ginv1=Ginv(marker.file=G.marker,ped.file=G.pedigree,aped.rowNames=gpnames,Goptions=1)

gblup<-asreml(t1~1+Site,random=~ vm(ID,Ginv1), data=G.data)

summary(gblup)$varcomp
c<-coef(gblup)$random

#### for user

df<-AFfR::read.file("data.csv",T,",")
ped<-AFfR::read.file("pedigree.csv",T,",")

##2.1 working for asreml V3
library(asreml) # for V3

pedinv<-asreml.Ainverse(ped)$ginv
apnames<-attributes(pedinv)$rowNames

path="G:/Users/yzhlin/Desktop/GBLUP"
Ginv1=Ginv(marker.file="Genotype.csv",ped.file="pedigree.csv",
           aped.rowNames=apnames,path=path,Goptions=1)


## GBLUP
df.asr<-asreml(t1~1+Site,random=~ giv(ID),
               ginverse=list(ID=Ginv1),data=df)

## ABLUP
df2.asr<-asreml(t1~1+Site,random=~ ped(ID),
                ginverse=list(ID=pedinv),data=df)

summary(df.asr)$varcomp
summary(df2.asr)$varcomp


##2.2 working for asreml V4
library(asreml) # for V4

pedinv<-ainverse(ped)
apnames<-attributes(pedinv)$rowNames

path="G:/Users/yzhlin/Desktop/GBLUP"
Ginv1=Ginv(marker.file="Genotype.csv",ped.file="pedigree.csv",
           aped.rowNames=apnames,path=path,Goptions=1)

## GBLUP
df.asr<-asreml(t1~1+Site,random=~ vm(ID,Ginv1),data=df)

## ABLUP
```

```
df2.asr<-asreml(t1~1+Site,random=~ vm(ID,pedinv),data=df)

summary(df.asr)$varcomp
summary(df2.asr)$varcomp

## write.table(Ginv1,"Ginv.giv",sep=" ", row.names=F, col.names=F,quote = F)
```

heatmap1 *Create a Heatmap*

Description

Function creates a correlation heatmap using ggplot2 given a data.frame

Usage

```
heatmap1(
  df,
  type = "matrix",
  df.label = NULL,
  gtitle = NULL,
  Nbreaks = NULL,
  Sig = FALSE,
  order = FALSE,
  diagp = FALSE,
  theme.1 = NULL,
  data.only = FALSE
)
```

Arguments

`df`	A data.frame or matrix containing only numeric data.
`type`	Identify df to be 'matrix'(default) or 'data'.
`df.label`	A matrix for heatmap labels.
`gtitle`	guide or legend title.
`Nbreaks`	A number controls legend breaks.
`Sig`	Logical, if TRUE put pvalue and sig level for heatmap labels.
`data.only`	Logical, if TRUE returns correlation and pvalue.

IC *Computes AIC and BIC for a model.*

Description

Computes Akiake and Bayesian (Schwarz) Information Criteria for a model. The function IC is provided for backwards compatibility.

Usage

```
## S3 method for class 'esR'
IC(object)

## S3 method for class 'asreml'
IC(object)

## S3 method for class 'remlf90'
IC(object)

## S3 method for class 'mmer'
IC(object)
```

Arguments

object	An asreml object resulting from the fitting of a model using REML.

Value

A data frame containing the Residual degrees of freedom, AIC, BIC, log of the REML value, and -2 times of log of the REML value.

Examples

```
## Not run:

library(asreml) # V4 or V3
library(AFfR)

data(PrSpa)
df<-PrSpa

fm.asr<-asreml(h5~1+Rep, random=~Fam, subset=Spacing=='3',data=df)

IC(fm.asr)


## End(Not run)
```

`mc.run` *multi-core run in R*

Description

multi-core run in R

Details

This page would show how to run with multi-core in R.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
## Not run:
code<-c('600519.ss','601398.ss')

# stock.get is the main function
# using lapply and stock.get
mm<-lapply(code, stock.get)

# using  stock.get with multi-core
library(doParallel)
cl <- makePSOCKcluster(detectCores()-1)
registerDoParallel(cl)

## multi-core run here
mm1<-foreach(code=code) %dopar% stock.get(code)

stopCluster(cl)

## End(Not run)
```

`MET` *multi-environment trial dataset*

Description

A dataset of one plant in multi-environment trials(MET)

Usage

```
data(MET)
```

Format

one dataset with 9 variables

Genotype levels with 36

Loc levels with 6

Loc2 levels with 6

Row levels with 9

Col levels with 18

Rep Replicate,levels with 3

Block levels with 6

Plot levels with 6

yield a numeric vector

`met.biplot` *Biplot asreml-MET results.*

Description

`met.biplot` This function biplots MET factor analytic results from asreml to find the relation of trial sites and the best variety suitable to trial sites.

Usage

```
met.biplot(object, dSco.u=NULL,dLam.u=NULL)

## S3 method for class 'asreml'
met.biplot(object, dSco.u = NULL, dLam.u = NULL)
```

Arguments

`object`	Asreml factor analytic results for MET.
`dSco.u`	Least score of Variety breeding value.
`dLam.u`	Least distance from center.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
library(AFfR)
data(MET)
MET$yield<-0.01*MET$yield

## for ASReml-R V3.0
library(asreml)
met.asr<-asreml(yield~Loc, random=~ Genotype:fa(Loc,2),
                rcov=~ at(Loc):ar1(Col):ar1(Row),
                data=MET, maxiter=50)

met.biplot(met.asr)
met.biplot(met.asr,dSco.u=1.8,dLam.u=1.0)


## for ASReml-R V4.1
library(asreml)
met.asr2<-asreml(yield~Loc, random=~ Genotype:fa(Loc,2),
                residual=~ dsum(~ar1(Col):ar1(Row)|Loc),
                data=MET, maxiter=50)

met.biplot(met.asr2)
met.biplot(met.asr2,dSco.u=1.8,dLam.u=1.0)
```

met.corr *asreml-MET corr matrix.*

Description

met.corr This function calculate var/cov/corr from asreml MET factor analytic results to further research the relation of trial sites.

Usage

```
met.corr(object, ...)

## S3 method for class 'asreml'
met.corr(
  object,
  aimS = NULL,
  rotate = FALSE,
  resOUT = FALSE,
  kn = NULL,
  plot = FALSE,
  plg = "S",
  dmethod = "manhattan"
)
```

Arguments

`object` Asreml factor analytic results for MET, such as met.asr.

`aimS` Specify the aim location parts of asreml object to count corr matrix.

`rotate` Rotate the factor loadings, FALSE(default).

`resOUT` Return 'Cov/Var/Corr matrix', FALSE(default).

`kn` Site cluster group numbers, 3(default).

`plot` Plotting site cluster, FALSE(default).

`plg` Adding labels before site, "S"(default).

`dmethod` The distance measured method for site cluster, "manhattan"(default), more details see amap::hcluster.

Value

return result directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
library(AFfR)
data(MET)
MET$yield<-0.01*MET$yield

## for ASReml-R V3.0
library(asreml)

met.asr1<-asreml(yield˜Loc, random=˜ Genotype:fa(Loc,2),
                rcov=˜ at(Loc):ar1(Col):ar1(Row),
                data=MET, maxiter=50)

met.corr(met.asr1)
met.corr(met.asr1,rotate=TRUE)

## for ASReml-R V4.1
library(asreml)

met.asr2<-asreml(yield˜Loc, random=˜ Genotype:fa(Loc,2),
                residual=˜ dsum(˜ar1(Col):ar1(Row)|Loc),
                data=MET, maxiter=50)

met.corr(met.asr2)
met.corr(met.asr2,rotate=TRUE)


# for complex met-fa model
MET$yield=MET$yield/100
```

```
set.seed(12345)
MET$hh1<-runif(648,10,50)

asreml.options(ai.sing=TRUE)
met.asr3<-asreml(cbind(yield,hh1)~trait+trait:Loc,
             random=~at(trait):us(trait):fa(Loc,1):Genotype,
             residual=~units:us(trait),
             #residual=~dsum(~units:diag(trait)|Loc),
             maxit=10,data=MET)

met.corr(met.asr3,aimS=1:15)
```

met.plot *Plot asreml-MET data.*

Description

met.plot This function plots MET data for further factor analytic by asreml to find the relation of trial sites, etc.

Usage

```
met.plot(object, plot.title = NULL)
```

Arguments

object	MET data.
plot.title	MET plot title.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
library(AFfR)
data(MET)
# example 1
# variable order: genotype,yield,site,row,col
MET2<-MET[,c(1,9,2,4:5)]
met.plot(MET2)

# example 2
MET3<-MET[,c(1,9,2,4:7)] # add variable order on MET2: Rep, Block
met.plot(MET3,"My met trials")
```

model.comp *Model comparison for asreml.*

Description

`model.comp` This function would compare models with different random structure under the same fixed factors.

Usage

```
model.comp(mulM,LRT=NULL,rdDF=NULL)
```

Arguments

`mulM`	A vector with more than 2 asreml results, such as ”c(m1,m2,m3,m4)”.
`LRT`	Value TRUE for Likelihood ratio test (LRT), default (FALSE) for no LRT.
`rdDF`	Value TRUE to control Ddf minus 0.5, default FALSE for LRT with Ddf. If TRUE, Ddf would minus 0.5, ie, Ddf=Ddf-0.5.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

Examples

```
library(AFfR)
df<-PrSpa

library(asreml) # for V4.1 or V3.0

# when for V3.0,  use 'rcov' not 'residual'.
fm1a<-asreml(cbind(dj,h5)~ trait+trait:Rep,
            random=~ us(trait):Fam,
            residual=~units:us(trait),
            subset=Spacing=='3',data=df,maxit=40)

fm1b<-asreml(cbind(dj,h5)~ trait+trait:Rep,
            random=~ diag(trait):Fam,
            residual=~units:us(trait),
            subset=Spacing=='3',data=df,maxit=40)

fm1c<-asreml(cbind(dj,h5)~ trait+trait:Rep,
            random=~ diag(trait):Fam,
            residual=~units:diag(trait),
            subset=Spacing=='3',data=df,maxit=40)
```

```
#####   model comparison    #####
model.comp(mulM=c(fm1a,fm1b))
model.comp(mulM=c(fm1a,fm1b),LRT=TRUE)
model.comp(mulM=c(fm1a,fm1b),LRT=TRUE,rdDF=TRUE)

model.comp(mulM=c(fm1a,fm1b,fm1c))
model.comp(mulM=c(fm1a,fm1b,fm1c),LRT=TRUE,rdDF=TRUE)
```

PC.res *output PCs for AMMI analysis*

Description

PC.res This function outputs PCs for AMMI analysis.

Usage

```
PC.res(model, pcN = 2)
```

Arguments

model	an AMMI object.
pcN	keeping pc number (default, 2).

Examples

```
library(AFfR)
data(plrv,package='agricolae')
names(plrv) #str(plrv)

library(agricolae)
model<- with(plrv,AMMI(ENV=Locality,GEN=Genotype,REP=Rep,
                       Y=Yield,console=F,PC=T))

model$ANOVA # for aov
model$analysis # for pc

PCs.aov<-PC.res(model,2) # for GEI aov
PCs.aov
```

pin *Count error for h2 and corr.*

Description

pin This function counts standard error(se) for heritability(h2) and corr value and also outputs significent level for corr value in asreml and breedR package.

Usage

```
## S3 method for class 'esR'
pin(object)

## S3 method for class 'asreml'
pin(
  object,
  formula = NULL,
  signif = FALSE,
  corM = FALSE,
  digit = 3,
  Rres = FALSE
)

## S3 method for class 'remlf90'
pin(object, formula = NULL, signif = FALSE, digit = 3, Rres = FALSE)

## S3 method for class 'mmer'
pin(object, formula = NULL, signif = FALSE, digit = 3, Rres = FALSE)
```

Arguments

`object`	asreml or breedR results.
`formula`	formula for h2 or corr.
`digit`	Index for decimal number, 3(default).
`signif`	Index to output signif levels, FALSE(default) for non-signif.
`Rres`	Index(TRUE) to restore results, FALSE(default) not.
`corM`	Corr model, FALSE(default).

Details

Count error for h2 and corr value, also outputs significent level.

Value

the result is returned directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

See Also

pin4, pin.batch, pin2.batch, repin, met.corr,asreml.batch, asreml.batch2,update, update2, coef2,IC,Var

Examples

```
## Not run:
## working for breedR package
library(breedR)
library(AFfR)

data(globulus)
res.animal <- remlf90(fixed = phe_X ~ 1,
                      random = ~ gg,
                      genetic = list(model = 'add_animal',
                      pedigree = globulus[, 1:3],
                      id = 'self'),
                      data = globulus)

pin(res.animal,h2~V2/(V1+V2+V3))

## End(Not run)

## Not run:
## working for asreml or asreml4 package
library(AFfR)
data(PrSpa)
df<-PrSpa

## when works for ASReml-R
library(asreml) # V3.0 or V4.1
# library(asreml4) #V4.0

# exmaple 1.1 single trait model
fm1<-asreml(h5~1+Rep, random=~Fam,
             subset=Spacing=='3',data=df)

summary(fm)$varcomp[,1:3]

pin(fm1, h2 ~4*V1/(V1+V2))
pin(fm1, h2 ~4*V1/(V1+V2),Rdf=TRUE)

# exmaple 1.2 us model
fm2<-asreml(cbind(dj,h5)~ trait+trait:Rep,
            random=~ us(trait):Fam,
            rcov=~units:us(trait),      # for V3
            #residual=~units:us(trait), # for V4
            subset=Spacing=='3',data=df,maxit=40)

summary(fm2)$varcomp[,1:3]

pin(fm2, h2_A ~ 4 * V1/(V1+V5)) # heritability for trait A
pin(fm2, h2_B ~ 4 * V3/(V3+V7)) # heritability for trait B

# genetic corr
pin(fm2, gCORR ~ V2/sqrt(V1*V3),signif=TRUE)

# phenotype corr
pin(fm2, pCORR ~ (V2+V6)/sqrt((V1+V5)*(V3+V7)),signif=TRUE)

## run batch pin()
bp<-c(h2_A ~ 4 * V1/(V1+V5),
      h2_B ~ 4 * V3/(V3+V7),
      gCORR ~ V2/sqrt(V1*V3),
      eCORR ~ V6/sqrt(V5*V7),
      pCORR ~ (V2+V6)/sqrt((V1+V5)*(V3+V7)))

pin.batch(fm2b.asr,bp,digit=5)
pin.batch(fm2b.asr,bp,digit=5,signif=T)


# exmaple 1.3 corr model
fm3<-asreml(cbind(dj,h3,h5)~ trait+trait:Rep,
            random=~ corgh(trait):Fam,
            rcov=~units:us(trait),      # for V3
            #residual=~units:us(trait), # for V4
            subset=Spacing=='3',data=df,maxit=40)

summary(fm3)$varcomp[,1:3]
pin(fm3,corM=TRUE)



## End(Not run)


## Not run:
# works for sommer
data(DT_btdata)
DT <- DT_btdata
mix4 <- mmer2(tarsus ~ sex, random = ~ dam + fosternest,
               data = DT)
summary(mix4)$var
pin(mix4, dam.prop ~ V1 / ( V1 + V2 + V3 ) )

## End(Not run)
```

---

pin.batch *Run pin() batch for ASReml-R or breedR*

Description

pin.batch This function will run batch pin() for ASReml-R or breedR.

Usage

```
pin.batch(object, mulp, digit = 4,
                        signif = FALSE,Rres=FALSE)

## S3 method for class 'asreml'
pin.batch(object, mulp, digit = 4, signif = FALSE, Rres = FALSE)

## S3 method for class 'remlf90'
pin.batch(object, mulp, digit = 4, signif = FALSE, Rres = FALSE)

## S3 method for class 'mmer'
pin.batch(object, mulp, digit = 4, signif = FALSE, Rres = FALSE)
```

Arguments

`object`	an asreml-run result.
`mulp`	batch formula for h2 or corr.
`digit`	Index for decimal number, 4(default).
`signif`	Index to output signif levels, F(default) for non-signif.
`Rres`	return results in a data frame.

Details

This function will run batch pin() for ASReml-R or breedR.

Value

this returned results directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

See Also

See Also as `pin` `pin4` `pin2.batch`

Examples

```
## Not run:
library(AFfR)
data(PrSpa)
df<-PrSpa

## batch formula
bp<-c(h2_A ~ 4 * V1/(V1+V5),
      h2_B ~ 4 * V3/(V3+V7),
      gCORR ~ V2/sqrt(V1*V3),
      eCORR ~ V6/sqrt(V5*V7),
      pCORR ~ (V2+V6)/sqrt((V1+V5)*(V3+V7)))

## when works for ASReml-R
```

```
library(asreml) # for V4.1 or V3.0

# when for V3.0, rcov=~units:us(trait)
fm2a.asr<-asreml(cbind(dj,h5)~ trait+trait:Rep,
            random=~ us(trait):Fam,
            residual=~units:us(trait),
            subset=Spacing=='3',data=df,maxit=30)


pin.batch(fm2a.asr,bp,digit=5)
pin.batch(fm2a.asr,bp,digit=5,signif=T)

# if having run two models object with 'fm2' and 'fm1'.
#pin2.batch(fm2,fm1,mulp=bp,signif=F,digit=7)
#pin22.batch(fm2,fm1,mulp=bp,signif=F,digit=7)


## when works for breedR
library(breedR)

df1<-subset(df,Spacing=='3')
fm2a.reml<-remlf90(cbind(dj,h5)~ Rep,
            random=~ Fam, data=df1)

Var(fm2a.reml)
## batch formula
bp1<-c(h2_A ~ 4 * V1/(V1+V4),
      h2_B ~ 4 * V3/(V3+V6),
      gCORR ~ V2/sqrt(V1*V3),
      eCORR ~ V5/sqrt(V4*V6),
      pCORR ~ (V2+V5)/sqrt((V1+V4)*(V3+V6)))

pin.batch(fm2a.reml,bp1,digit=5)
pin.batch(fm2a.reml,bp1,digit=5,signif=T)


## End(Not run)

## Not run:
# works for sommer
data(DT_btdata)
DT <- DT_btdata
mix4 <- mmer2(tarsus ~ sex, random = ~ dam + fosternest,
               data = DT)
summary(mix4)$var
pin(mix4, dam.prop ~ V1 / ( V1 + V2 + V3 ) )

pin.batch(mix4, mulp=c(dp ~ V1 / ( V1 + V2 + V3 ),
                       dp2 ~ 4*V1 / ( V1 + V2 + V3 )))
pin.batch(mix4, mulp=c(dp ~ V1 / ( V1 + V2 + V3 ),
                       dp2 ~ 4*V1 / ( V1 + V2 + V3 )),
                       signif=TRUE)

## End(Not run)
```

pin22.batch.asreml *Run pin batch for ASReml-R*

Description

pin2.batch This function will run batch pin for ASReml-R.

Usage

```
## S3 method for class 'asreml'
pin22.batch(..., mulp, signif = F, digit = 4, Rres = F)

pin2.batch(..., mulp, digit = 4,
                  signif = FALSE,Rres=FALSE)

## S3 method for class 'asreml'
pin2.batch(..., mulp, digit = 4, signif = FALSE, Rres = FALSE)
```

Arguments

...	A list of asreml results,used by 'm1,m2,...'.
mulp	batch formula for h2 or corr.
signif	Index to output signif levels, F(default) for non-signif.
digit	Index for decimal number, 4(default).
Rres	return results in a list.

Details

This function will run batch pin() for ASReml-R.

Value

this returned results directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

See Also

pin pin4 pin.batch

Examples

```
## Not run:

library(AFfR)

data(PrSpa)
df<-PrSpa

## batch formula
bp<-c(h2 ~ 4 * V1/(V1+V2), h2_B ~ V1/(V1+V2/4))

library(asreml) # for V4.1 or V3.0

fm1.asr<-asreml(h5~ 1+Rep,
            random=~ Fam,
            subset=Spacing=='3',data=df,maxit=30)

fm2.asr<-update(fm1.asr,fixed=h3~ 1+Rep)

pin2.batch(fm1.asr,fm2.asr,bp,digit=5)
pin2.batch(fm1.asr,fm2.asr,bp,digit=5,signif=T)
pin22.batch(fm1.asr,fm2.asr,bp,digit=5,signif=T)


## End(Not run)
```

---

pin4 *Count standard error for h2 and corr for ASReml V4.1*

Description

`pin4` This function will count standard error for h2 and corr for ASReml V4.1.

Usage

```
pin4(object, ...)

## S3 method for class 'asreml'
pin4(
  object,
  formula = NULL,
  signif = FALSE,
  corM = FALSE,
  digit = 3,
  Rres = FALSE
)
```

Arguments

`object`	asreml or breedR results.
`formula`	formula for h2 or corr.

`signif` Index to output signif levels, F(default) for non-signif.

`corM` Corr model, FALSE(default).

`digit` Index for decimal number, 4(default).

Details

This function will count standard error for h2 and corr for ASReml V4.1.

Value

this returned results directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016

See Also

See Also as `pin`

Examples

```
## Not run:
## when works for ASReml-R V4.1
library(asreml)
library(AFfR)

data(PrSpa)
df<-PrSpa


# exmaple 1 single trait model, default gamma parameterization.
fm1b<-asreml(h5˜1+Rep, random=˜Fam, subset=Spacing=='3',data=df)

summary(fm)$varcomp[,1:3]

pin4(fm1b, h2 ˜4*V1/(V1+V2))

# The same model with fm1b, but with sigma parameterization.
fm1c<-asreml(h5˜1+Rep, random=˜Fam,
               residual=˜idv(units),
               subset=Spacing=='3',data=df)

summary(fm1c)$varcomp[,1:3]

pin4(fm1c, h2 ˜4*V1/(V1+V3))

# exmaple 2 us model
fm2b<-asreml(cbind(h3,h5)˜ trait+trait:Rep,
            random=˜ us(trait):Fam,
            residual=˜units:us(trait),
            subset=Spacing=='3',data=df,maxit=40)

summary(fm2b)$varcomp[,1:3]

# heritability for trait A
pin4(fm2b, h2_A ~ 4 * V1/(V1+V5))

# heritability for trait B
pin4(fm2b, h2_B ~ 4 * V3/(V3+V7))

# genetic corr
pin4(fm2b, gCORR ~ V2/sqrt(V1*V3),signif=TRUE)

# phenotype corr
pin4(fm2b, pCORR ~ (V2+V6)/sqrt((V1+V5)*(V3+V7)),signif=TRUE)

# exmaple 3 corr model
fm3b<-asreml(cbind(h3,h4,h5)~ trait+trait:Rep,
            random=~ corgh(trait):Fam,
            residual=~units:us(trait),
            subset=Spacing=='3',data=df,maxit=40)

summary(fm3b)$varcomp[,1:3]

pin4(fm3b,corM=TRUE)


## End(Not run)
```

plot1 *Plotting test trait's norm or multi-comparison*

Description

plot1 This function plots test trait's norm for breedR object or multi-comparisons for agricolae.

Usage

```
## S3 method for class 'esR'
plot1(object)

## S3 method for class 'remlf90'
plot1(object, mulT = FALSE)

## S3 method for class 'group'
plot1(object, x.lbls = NULL, y.lbls = NULL, y.zero = NULL, stdl = "sd")
```

Arguments

object an object of breedR result or multi-comparison.

mulT multi-trait model(default, FALSE).

x.lbls x axis label

`y.lbls`	y axis label
`y.zero`	y axis tick from zero (0, default) or other value.
`stdl`	add standard error(se) or standard deviance(sd, default).

Details

Test trait's norm for breedR object,similar to asreml.

Value

the result is returned directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

AFfR website:https://github.com/yzhlinscau/AFfR

Examples

```
## 1 working for agricolae package
library(AFfR)
library(agricolae)

data(sweetpotato)
model<-aov(yield~virus,data=sweetpotato)
comparison<- LSD.test(model,"virus",alpha=0.01)

# add sd
plot1(comparison,x.lbls="virus")

# add se
plot1(comparison,x.lbls="virus",stdl='se')

## Not run:
## 2 working for breedR package
library(breedR)
library(AFfR)

res.animal <- remlf90(fixed = phe_X ~ 1,
                      random = ~ gg,
                      genetic = list(model = 'add_animal',
                      pedigree = globulus[, 1:3],
                      id = 'self'),
                      data = globulus)

plot1(res.animal)

## End(Not run)
```

`predict2` *Predict for a LMM fitted with mmer*

Description

predict method for class "mmer".

Usage

```
predict2(object, ...)
```

Details

This function details see "predict.mmer" in sommer.

`PrSpa` *Tree spacing trial dataset*

Description

A dataset of spacing trials of Pinus radiata

Usage

```
data(PrSpa)
```

Format

one dataset with 13 variables

TreeID Tree Sample code

Spacing 3 levels

Rep Replication code

Fam Family code

Plot Plot code

dj a numeric vector

dm a numeric vector

wd a numeric vector

h1 a numeric vector

h2 a numeric vector

h3 a numeric vector

h4 a numeric vector

h5 a numeric vector

`read.example` *read file list*

Description

`read.example` This function read file list under one package, or sets working directory under package.

Usage

```
read.example(package, setpath = FALSE)
```

Arguments

`package`	package name.
`setpath`	Whether set working directory under package, FALSE(default).

Value

this returned a path or file list.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016 AFfR website:https://github.com/yzhlinscau/AFfR

See Also

See Also as `read.file`, `fdata`

Examples

```
library(AFfR)
# read file list under a package
AFfR::read.example(package = "AFfR")

# set working directory under a package
AFfR::read.example(package = "AFfR", setpath = TRUE)
getwd()
```

read.file *read file*

Description

read.file This function read file similar to asreml.read.table().

Usage

```
read.file(file, header = TRUE, sep = ",", dec = ".", ...)
```

Arguments

file	File name.
header	Whether file has header for Varialbes, TRUE(default).
sep	Field separator character, ','(default).
dec	Decimal points' character, '.'(default).
...	Further arguments to be passed to read.table.

Details

Count error for h2 and corr value, also outputs significent level.

Value

this returned a data.frame.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016 AFfR website:https://github.com/yzhlinscau/AFfR

See Also

See Also as read.example, fdata

Examples

```
library(AFfR)
AFfR::read.example(package = "AFfR", setpath = TRUE)
df<-AFfR::read.file(file="fm.csv", header=TRUE)
names(df)
```

repin *rerun pin for asreml or breedR batch result*

Description

recalculate new parameters by `repin` for asreml or breedR batch results.

Usage

```
repin(object, mulp)

## S3 method for class 'asreml.batch'
repin(object, mulp)

## S3 method for class 'breedR.batch'
repin(object, mulp)

## S3 method for class 'sommer.batch'
repin(object, mulp)
```

Arguments

`object`	an object of batch results from `asreml.batch` or `breedR.batch`.
`mulp`	new formula for parameters to be calculated,used as 'c()'.

Value

the results return directly.

See Also

`update` `asreml.batch`

Examples

```

## Not run:

pkgs <- c('asreml','AFfR','breedR')
lapply(pkgs,library,character.only=TRUE)

df1<-subset(PrSpa,Spacing=='3')
mulp1 <- c(h2 ~ 4 * V1/(V1+V2))

# works for asreml
run1 <- asreml.batch(data=df1,factorN=1:5,traitN=c(9:13),
                     FMod=y~1+Rep+Plot,
                     RMod=~Fam,
                     all.result=TRUE,
                     # family=asreml.gaussian(), # for V3
                     # family=asr_gaussian(),    # for V4
                     mulp=mulp1)
```

```
# works for breedR
run1 <- breedR.batch(data=df1,factorN=1:5,traitN=c(9:13),
                     FMod=y~1+Rep+Plot,
                     RMod=~Fam,
                     all.result=TRUE,
                     mulp=mulp1)

repin(run1,mulp=c(h2 ~ 4 * V1/(V1+V2),
                  H2 ~  V1/(V1+V2/4)))



## End(Not run)

## Not run:
# works for sommer
data(DT_cpdata,package='sommer')
DT <- DT_cpdata
GT <- GT_cpdata
A <- sommer::A.mat(GT)

library(sommer)
run1 <- sommer.batch(data=DT,factorN=c(1,9:10),traitN=c(6:7),
             FMod=y~ 1,RMod=~ vs(id,Gu=A)+Rowf + Colf,
             EMod=~units,all.result=T,
             mulp=c(h2~V1/(V1+V4)))

repin(run1,mulp=c(h2 ~ 4 * V1/(V1+V2),
                 H2 ~  V1/(V1+V2/4)))

## End(Not run)
```

sp *sp*

Description

An example dataset of spatial analysis.

Usage

```
data(sp)
```

Format

one dataset with 10 variables

Tree Tree ID

Mum Mum ID, 34 levels

Dad Dad ID, 32 levels

Fam Family ID, 42 levels

Row Row number

Col Column number

Rep Replication level with 7

Plot Plot level with 5

dbh10 a numeric vector

t2 a numeric vector

`spd.plot` *Plot spatial data or Variogram.*

Description

`spd.plot` This function plots spatial data or Variogram.

Usage

```
spd.plot(object,type="data",p.lbls=NULL,key.unit=NULL,
                x.unit=NULL,y.unit=NULL,na=NULL,
                color.p=NULL,mtitle=NULL)
```

Arguments

`object`	Aim dataset.
`type`	Type of dataset, default value is "data", when "Variogram" for Variogram.plot in spatial analysis in ASReml-R.
`p.lbls`	Extra labels in figure title.
`key.unit`	The unit of key, default value is 1.
`x.unit`	Axis x least unit, default value is 1.
`y.unit`	Axis y least unit, default value is 1.
`na`	Transform NA to 0(na=0) or keep NA (default).
`color.p`	Parameters of the colors for figures, default value is terrain.colors, it could be rainbow, heat.colors, cm.colors and topo.colors.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016 AFfR website:https://github.com/yzhlinscau/AFfR

Examples

```
library(AFfR)

######## example 1 plot regular spatial data
data(barley)

aim.trait<-subset(barley,select=c(Row,Column,yield))
spd.plot(aim.trait)
spd.plot(aim.trait,color.p=topo.colors)
spd.plot(aim.trait,key.unit="Kg")
spd.plot(aim.trait,p.lbls="barley",x.unit=2,y.unit=1)

#AR1XAR1--asreml V3.0
barley1.asr<-asreml(yield~Variety, rcov =~ ar1(Row):ar1(Column), data=barley)

summary(barley1.asr)$Varcomp
plot(Variogram(barley1.asr),main="M1")

aa=Variogram(barley1.asr)
spd.plot(aa,type="Variogram",color.p=topo.colors)

######## example 2 plot spatial data with NA's
data(ir.sp)

ir.sp2<-ir.sp[,5:16] # order: Row,Col,h05,cw05,...
#ir.sp2<-subset(ir.sp,select=c(Row,Col,h05,cw05))

sp1<-ir2r.sp(ir.sp2,row.max=10,col.max=20)

aim.trait=subset(sp1,select=c(Row,Col,d10))
spd.plot(aim.trait,key.unit="cm")
spd.plot(aim.trait,color.p=topo.colors,na=0)
spd.plot(aim.trait,na=0,x.unit=3)
```

stock.fun *Simple trading method for stock.*

Description

stock.fun This function would first get stock data by its code, then runs some simple trading methods. It also works for stock data with 'xts' format data directly. Note: all analysis focus on closed price.

Usage

```
stock.fun(STOCK,
                  Sdate=NULL,Edate=NULL,
                  title="Stock",breaks='1 month',
                  trade.test=FALSE,
                  capital=100000,fixMoney=10000,...)
```

Arguments

`STOCK`	'xts' format stock data.
`Sdate`	The starting date for stock data .
`Edate`	The end date for stock data.
`title`	Title for plots.
`breaks`	Time breaks for plots.
`trade.test`	Runing a mock trading for stock, F(default).
`capital`	total cash for mock trading,100000(default).
`fixMoney`	fixed money for each buy, 10000(default).
`...`	additional parameters.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016
AFfR website:https://github.com/yzhlinscau/AFfR

See Also

See Also as `stock.get`

Examples

```
## Not run:
library(AFfR4)

##### simple analysis examples

# one stock from 2018-01-01
MT<-stock.get('600519.ss')
mm1<-stock.fun(STOCK=MT) # return list

names(mm1)
mm1$fig7
mm1$bsdat2

# two or more stock from 2018-01-01
code<-c('600519.ss','601398.ss')

mm<-lapply(code, stock.get) # return list
sn<-c('MT',"ICBC")
names(mm)<-sn # renames

mm2<-lapply(mm,stock.fun) # return list
mm2$MT$fig1
mm2$MT$bsdat2

mm2$ICBC$fig7
mm2$ICBC$bsdat2
```

```
# plot fig7 together
res3<-refm.tdat(mm2,sname=sn)
drawPoint2(res3)


## End(Not run)
```

stock.get *get stock dataset*

Description

`stock.get` This function will get stock dataset.

Usage

```
stock.get(code, Sdate = NULL, Edate = NULL, Sdate0 = FALSE, rename = TRUE)
```

Arguments

`Sdate`	The starting date for stock data,'2018-01-01'(default).
`Edate`	The end date for stock data.
`Sdate0`	The earliest date for stock data,FALSE(default).
`rename`	rename the variable names for stock, TRUE(default).

Details

This function will get stock dataset with some conditions.

Value

this returned 'xts' format stock data.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

Yuanzhen Lin. R & ASReml-R Statistics. China Forestry Publishing House. 2016
AFfR website:https://github.com/yzhlinscau/AFfR

See Also

See Also as `stock.fun`

Examples

```
## Not run:
#### get one stock data

# from the earliest date to special date
MT<-stock.get('600519.ss',Edate='2008-03-01',Sdate0 = T)

# data from 2018-01-01 to now
MT<-stock.get('600519.ss')

# data between two special dates
MT<-stock.get('600519.ss',Sdate='2016-01-01',Edate='2008-03-01')

head(MT)

#### get more than one stock
code<-c('600519.ss','601398.ss')

mm<-lapply(code,stock.get)
names(mm)<-c('MT',"ICBC")

tail(mm$MT)
tail(mm$ICBC)

## End(Not run)
```

`update` *rerun new batch from old batch result in asreml*

Description

rerun new batch analysis by `update2` for asreml batch results.

Usage

```
update(object, ...)

## S3 method for class 'asreml.batch'
update(object, FMod = NULL, RMod = NULL, EMod = NULL, maxit = 30)

## S3 method for class 'breedR.batch'
update(
  object,
  FMod = NULL,
  RMod = NULL,
  genetic = NULL,
  spatial = NULL,
  generic = NULL,
  weights = NULL,
  method = "ai"
)
```

```
## S3 method for class 'sommer.batch'
update(object, FMod = NULL, RMod = NULL, EMod = NULL, maxit = 30)
```

Arguments

`object` a object of asreml batch results from `asreml.batch` or `asreml.batch2`.

`FMod` new Fixed mode.

`RMod` new Randomed variance structure (G structure).

`EMod` new Error variance structure (R) for multi-trait model.

`maxit` Maximum number of iterations, 30(default).

Value

the results will return directly for variance compnents and a list for `repin`.

See Also

`asreml.batch` `asreml.batch2` `repin`

Examples

```
## Not run:

pkgs <- c('asreml','AFfR')
lapply(pkgs,library,character.only=TRUE)

data(PrSpa)
df1<-subset(PrSpa,Spacing==3)

mulp1 <- c(h2 ˜ 4 * V1/(V1+V2))

run1 <- asreml.batch2(data=df1,factorN=1:5,traitN=c(9:13),
                     FMod=y˜1+Rep+Plot,
                     RMod=˜Fam,
                     all.result=TRUE,
                     # family=asreml.gaussian(), # for V3
                     # family=asr_gaussian(),    # for V4
                     mulp=mulp1)

repin(run1,mulp=c(h2 ˜ 4 * V1/(V1+V2),
                        H2 ˜  V1/(V1+V2/4)))

## re-run new batch
run1a<-update(run1,FMod=y˜1+Rep)

repin(run1a,mulp=c(h2 ˜ 4 * V1/(V1+V2),
                         H2 ˜  V1/(V1+V2/4)))

## End(Not run)
```

update2 *rerun new batch from old batch result in asreml*

Description

rerun new batch analysis by `update` for asreml batch results.

Usage

```
update2(object, ...)

## S3 method for class 'asreml.batch'
update2(object, FMod = NULL, RMod = NULL, EMod = NULL, maxit = 30)
```

Arguments

`object`	a object of asreml batch results from `asreml.batch`.
`FMod`	new Fixed mode.
`RMod`	new Randomed variance structure (G structure).
`EMod`	new Error variance structure (R) for multi-trait model.
`maxit`	Maximum number of iterations, 30(default).

Value

the results will return directly for variance compnents and a list for `repin`.

See Also

`asreml.batch` `repin`

Examples

```

## Not run:

pkgs <- c('asreml','AFfR')
lapply(pkgs,library,character.only=TRUE)

data(PrSpa)
df1<-subset(PrSpa,Spacing==3)

mulp1 <- c(h2 ~ 4 * V1/(V1+V2))

run1 <- asreml.batch(data=df1,factorN=1:5,traitN=c(9:13),
                     FMod=y~1+Rep+Plot,
                     RMod=~Fam,
                     all.result=TRUE,
                     # family=asreml.gaussian(), # for V3
                     # family=asr_gaussian(),    # for V4
                     mulp=mulp1)

repin(run1,mulp=c(h2 ~ 4 * V1/(V1+V2),
                        H2 ~  V1/(V1+V2/4)))

## re-run new batch
run1a<-update2(run1,FMod=y~1+Rep)

repin(run1a,mulp=c(h2 ~ 4 * V1/(V1+V2),
                         H2 ~  V1/(V1+V2/4)))

## End(Not run)
```

Var *Output Variance components for R packages*

Description

Var This function output Variance components for R packages.

Usage

```
## S3 method for class 'lme'
Var(object)

## S3 method for class 'lmerMod'
Var(object)

## S3 method for class 'asreml'
Var(object)

## S3 method for class 'remlf90'
Var(object, mulT = FALSE)

## S3 method for class 'esR'
Var(object)
```

Arguments

object	an object of mixed model results from R packages.
mulT	multi-trait model(default, FALSE).

Details

Output Variance component for mixed model results from R packages.

Value

the result is returned directly.

Author(s)

Yuanzhen Lin ¡yzhlinscau@163.com¿

References

AFfR website:https://github.com/yzhlinscau/AFfR

Examples

```
library(AFfR)

## 00 data
data(butron.maize,package='agridat')
df<-butron.maize

## 01 nlme package
library(nlme) # V3.1-131

nlm<-lme(yield~1+env,random=~1|male/female,
               na.action='na.omit',
               data=df)

Var(nlm)

## 02 lme4 package
library(lme4) # V1.1-17

lme<-lmer(yield~1+env+(1|male)+(1|female),
               data=df)

Var(lme)

 ## Not run:
## 03 breedR package
library(breedR) # V0.12-1

bdR <- remlf90(fixed = yield~1+env,
                      random = ~ male+female,
                      data=df)

Var(bdR)


## 04 asreml package
library(asreml) #V3.0

asr <- asreml(fixed = yield~1+env,
                      random = ~ male+female,
                      na.method.X='include',
                      data=df)

Var(asr)

##### special for breedR
library(breedR)
library(AFfR)

data(globulus)
res.animal <- remlf90(fixed = phe_X ~ 1,
                      random = ~ gg,
                      genetic = list(model = 'add_animal',
                      pedigree = globulus[, 1:3],
                      id = 'self'),
                      data = globulus)
Var(res.animal)

## End(Not run)
```

Index