Package ‘qtlmt’

October 13, 2022

Version 0.1-6

Date 2017-07-20

Title Tools for Mapping Multiple Complex Traits

Author Riyan Cheng [aut, cre]

Maintainer Riyan Cheng <riyancheng@hotmail.com>

Description Provides tools for joint analysis of multiple traits in a backcross (BC) or recombinant inbred lines (RIL) population. It can be used to select an optimal subset of traits for multiple-trait mapping, analyze multiple traits via the SURE model, which can associate different QTL with different traits, and perform multiple-trait composite multiple-interval mapping.

Depends R (>= 2.10)

Imports

LazyLoad yes

LazyData no

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-07-25 20:31:45 UTC

R topics documented:

`dataSets` *Data sets*

Description

A collection of data sets for demonstration.

Usage

```
data(etrait)
```

`mAdd1` *Add or drop all possible terms*

Description

Add or drop all possible terms to or from a multivariate multiple regression model.

Usage

```
mAdd1(object, scope, test = c("none","Chisq","F"), k = 0, ...)

mDrop1(object, scope, test = c("none","Chisq","F"), k = 0, ...)
```

Arguments

`object`	initial model.
`scope`	a formula or a vector of variable names, giving lower/upper bound of the model.
`test`	if not "none", the test statistic and p-value will be included in the output.
`k`	penalty on a parameter in AIC.
`...`	additional arguments to `update`.

Value

An object summarizing the differences in fit between the models.

See Also

`add1` and `mStep`

Examples

```
data(etrait)
mdf<- data.frame(traits,markers)
## Not run:
mlm<- lm(cbind(T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16) ~
   m1 + m2 + m3 + m4 + m5, data=mdf)

up<- formula(paste("~", paste("m",1:15,collapse=" + ",sep="")))
oa<- mAdd1(mlm, scope=up, test="F", k=5, data=mdf)

lw<- formula(paste("~ ", paste("m",1:3,collapse=" + ",sep="")))
od<- mDrop1(mlm, scope=lw, test="F", k=5, data=mdf)

## End(Not run)
```

---

misFct *Miscellaneous functions*

Description

Functions that may be useful.

Usage

```
# create 'mpos' and 'dists'
gv2mpos(gmap,v)

# extract 'xid' on chromosome 'k'
xid1ch(mpos,v,k)

# extract covariate effects of a mtcmim object 'object'
xeff(a,xid)

# extract QTL effect from a mtcmim object 'object'
qeff(object)
```

Arguments

gmap	a genetic map, which is a data frame (chr=chromosome id, dist=genetic distance (cM) on the chromosome,...). The chromosome id should be an integer.
v	a list; v[[j]] indicates which x's in the model for y[,j].
mpos	a data frame (id=marker index, ch=chromosome id, m=marker index on the chromosome, dist=genetic position in cM on the chromosome). Chromosome id should be an integer.
k	which chromosome?
a	covariate effects of an mtcmim object.
xid	a list of length p, xid[[j]] specifies columns of x as covariates for y[,j] .
object	an mtcmim object.

See Also

mtcmim and sureEst.

mlogLik *Extract log-likelihood*

Description

Extract log-likelihood from the object of a multivariate multiple regression model.

Usage

```
mlogLik(object)
```

Arguments

object	object of a fitted multivariate multiple regression model.

Value

log-likelihood.

mStep *Model selection in multivariate multiple regression*

Description

Select a multivariate multiple regression model via model selection.

Usage

```
mStep(object, scope, direction=c("both","backward","forward"),
   trace=FALSE, keep=TRUE, steps=1000, k=2, ...)
```

Arguments

object	initial model in model search.
scope	a single formula, which provides 'upper', or a list containing components 'upper' and 'lower', both formulae; defines the lower and upper bound. See step.
direction	forward selection, backward elimination or stepwise.
trace	whether to track the process for monitoring purpose.
keep	whether to return the change of terms and related statistics.
steps	maximum number of search steps.
k	penalty on a parameter. The selection criterion is the known "AIC" if k = 2 and is "BIC" if k = log(n) where "n" is the sample size.
...	additional arguments to update.

Value

a list with components of a `lm` object plus ‘keep’ if required.

See Also

`mAdd1` and `mDrop1`

Examples

```
data(etrait)
mdf<- data.frame(traits,markers)
## Not run:
mlm<- lm(cbind(T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16) ~
   m1 + m2 + m3 + m4 + m5, data=mdf)

lw<- formula(paste("~ ", paste("m",1:3,collapse=" + ",sep="")))
up<- formula(paste("~", paste("m",1:15,collapse=" + ",sep="")))

ob<- mStep(mlm, scope=list(lower=lw), k=99, direction="backward", data=mdf)
of<- mStep(mlm, scope=list(upper=up), k=5, direction="forward", data=mdf)
o1<- mStep(mlm, scope=list(upper=up), k=5, direction="both", data=mdf)
o2<- mStep(o1, scope=list(upper=up), k=2, direction="forward", data=mdf)

## End(Not run)
```

`mtcmim` *MTCMIM*

Description

Multiple-trait composite multiple-interval mapping.

Usage

```
mtcmim(y, mpos, mdat, x, xid, dists, a, b, sigma, qtl=NULL,
   eps=NULL, win=Inf, range=0, pp=1, len=2, init=1,
   iter=10000,tol=1e-12)
```

Arguments

`y` a n by p matrix, whose columns are dependent variables.

`mpos` a data frame (id=marker index, ch=chromosome id, m=marker index on the chromosome, dist=genetic position in cM on the chromosome). Chromosome id should be an integer.

`mdat` a matrix of n rows; marker genotypes (1 or 0). Columns should correspond to markers in the order.

`x` covariates; n by m numerical matrix.

`xid` a list of length p, xid[[j]] specifies columns of x as covariates for y[,j] .

`dists` a data frame (ch=chromosome id, mid=marker index, d=genetic position in cM on the chromosome); specifies initial QTL locations. Chromosome id should be an integer.

`a` initial covariate effects including intercepts (if given).

`b` initial QTL effects (if given).

`sigma` initial residual variance-covariance (if given).

`qtl` a list of length p (if given); qtl[[j]] specifies qtls for y[,j], which are defined by rows of `dists`.

`eps` a data frame (y=which trait,q1=QTL one,q2=QTL two) (if not `NULL`); specifies epistatic terms.

`win` window width of search around existing QTL. Ignored if `range=0`.

`range` search range: genome-wide (0), the same chromosomes (-1).

`pp` mapping population: BC-1, RIL-selfing-2, RIL-brother-sister-mating-3.

`len` step length in search.

`init` whether `a`, `b` and `sigma` are used as initial values.

`iter` maximum number of iterations in a numerical process to estimate model parameters.

`tol` convergence tolerance.

Details

Given the covariates, the number of QTL and epistasis that are specified for each trait, this function searches for the optimal genomic locations of the QTL, and estimates the model parameters.

Value

a list with the following components:

`loglik` log-likelihood of the final model

`a` covariate effects

`b` QTL effects

`sigma` residual variance-covariance

`qtl` QTL for each trait

`eps` epistatic terms

`dists` QTL locations

Examples

```
data(etrait)
qtl<- vector("list",16); qtl[[1]]<- c(1,2)
eps<- data.frame(y=1,q1=1,q2=2)
dists<- dists[c(4,11),]
x<- mdat - 3/2
```

```
## Not run:
o<- mtcmim(traits, mpos, mdat, x, xid, dists, qtl=qtl, eps=eps,
   win=5, range=-1, pp=2, len=1)

## End(Not run)
```

mtcmimStep *MTCMIM model selection*

Description

Model selection for multiple-trait composite multiple-interval mapping.

Usage

```
mtcmimAdd1(object, y, x, xid, mpos, mdat, pp=1, len=1, type=1,
   iter=10000, tol=1e-12, ext=FALSE)

mtcmimDrop1(object, y, x, xid, mpos, mdat, pp=1, len=1, type=1,
   iter=10000, tol=1e-12, ext=FALSE)

mtcmimStep(object, y, x, xid, mpos, mdat, cv=0,
   direction=c("both","backward","forward"), pp=1, len=1,
   type=1, iter=10000, tol=1e-12, ext=FALSE)
```

Arguments

`object`	an object of class `mtcmim`.
`y`	a n by p matrix, whose columns are dependent variables.
`x`	covariates; n by m numerical matrix.
`xid`	a list of length p; xid[[j]] specifies columns of x as covariates for y[,j] .
`mpos`	a data frame (id=marker index, ch=chromosome id, m=marker index on the chromosome, dist=genetic position in cM on the chromosome). Chromosome id should be an integer.
`mdat`	a matrix of n rows; marker genotypes (1 or 0). columns should correspond to markers in the order.
`pp`	mapping population: BC-1, RIL-selfing-2, RIL-brother-sister-mating-3.
`len`	step length in search.
`type`	1 if traits can have the different sets of covariates and QTL, 2 if all have the same set of covariates and QTL.
`ext`	whether to perform an extensive search for an "optimal" model with the same number of QTL per phenotype.
`cv`	critical value used in the likelihood ratio test to determine adding/dropping a QTL.

`direction`	forward selection, backward elimination or both directions.
`iter`	maximum number of iterations in a numerical process to estimate model parameters.
`tol`	convergence tolerance.

Value

a list with the following components:

`loglik`	log-likelihood of the final model
`a`	covariate effects
`b`	QTL effects
`sigma`	residual variance-covariance
`qtl`	QTL for each trait
`dists`	QTL locations

Note

Currently, not able to include epistatic effects.

See Also

`mtcmim`

Examples

```
data(etrait)
y<- traits[,1:5]
qtl<- vector(”list”,5); qtl[[1]]<- c(1,2)
dists<- dists[c(4,11),]
x<- mdat - 3/2
## Not run:
o<- mtcmim(y, mpos, mdat, dists=dists, qtl=qtl, eps=NULL,
   win=5, range=-1, pp=2, len=1)
of<- mtcmimAdd1(o, y=y, mpos=mpos, mdat=mdat, pp=2, len=3)
os<- mtcmimStep(of, y=y, mpos=mpos, mdat=mdat, cv=25,
   direction=”both”, pp=2, len=3)

## End(Not run)
```

sureEps *Epistasis in a SURE model*

Description

Look for epistasis between existing QTL in a model via model selection.

Usage

```
sureEps(y, x, v, k, direction=c("both","backward","forward"),
   iter=10000, max.terms=200, steps=1000, tol=1e-12)
```

Arguments

y a n by p matrix, whose columns are dependent variables.

x a n by m matrix, whose columns are predictor variables

v a list; v[[j]] indicates which x's in the model for y[,j], between which possible epistasis is looked for.

k penalty, 0 if missing or <0.

direction forward selection, backward elimination or stepwise.

iter maximum number of iterations in a numerical process to estimate model parameters.

max.terms maximum number of terms in the final model.

steps maximum number of search steps.

tol convergence tolerance.

Value

associated traits, epistatic markers and p-values

See Also

surStep

Examples

```
data(etrait)
x<- as.matrix(mdat-1/2)
y<- as.matrix(traits)[,1:3]
v<- list()
upper<- list()
for(k in 1:ncol(y)){
   v[[k]]<- numeric(0)
   upper[[k]]<- 1:ncol(x)
}
## Not run:
```

```
o<- surStep(y, x, v=v, upper=upper, k=19, direction="both",
   iter=250, max.terms=250, steps=2000, tol=1e-12)
neps<- 6 # suppose there are 6 possible epistatic effects
oeps<- sureEps(y, x, o$v, k=qchisq(1-0.05/neps,1), direction="backward",
  iter=250, max.terms=200, steps=1000, tol=1e-12)

## End(Not run)
```

sureEst *SURE model parameter estimation*

Description

Estimate parameters in a SURE model.

Usage

```
sureEst(y, x, v, sigma, iter=10000, tol=1e-12)
```

Arguments

y	a n by p matrix, whose columns are dependent variables.
x	a n by m matrix, whose columns are predictor variables to select from.
v	a list; v[[j]] indicates which x's in the model for y[,j].
sigma	initial residual variance-covarance matrix (if given).
iter	maximum number of iterations in a numerical process to estimate model parameters.
tol	convergence tolerance.

Value

a list with the following components:

loglik	log-likelihood of the model
b	estimates of model coefficients
sigma	estimated residual variance-covariance
fitted.values	fitted mean values

Examples

```
data(etrait)
x<- as.matrix(mdat-1/2)
y<- as.matrix(traits)[,1:3]
v<- list(c(1,25,50),numeric(0),3)
## Not run:
o<- sureEst(y, x, v=v, iter=250, tol=1e-12)

## End(Not run)
```

sureStep *SURE model selection*

Description

Select a SURE model via model selection.

Usage

```
sureAdd1(object, y, x, range=NULL, iter=10000, tol=1e-12, ext=FALSE)

sureDrop1(object, y, x, range=NULL, iter=10000, tol=1e-12, ext=FALSE)

sureStep(object, y, x, cv, direction=c("both","backward","forward"),
   range=NULL, iter=10000, steps=1000, tol=1e-12, ext=FALSE)

surStep(y, x, v, lower, upper, k, direction=c("both","backward",
   "forward"), iter=10000, max.terms=200, steps=1000, tol=1e-12)
```

Arguments

object	initial model in model search; can be an object of sureEst.
y	a n by p matrix, whose columns are dependent variables.
x	a n by m matrix, whose columns are predictor variables to select from.
range	a list; range[[j]] indicates which x's (all if NULL) correspond to which y[,j].
cv	critical value used in the likelihood ratio test to determine adding/dropping a term.
v	a list; v[[j]] indicates which x's to start with for y[,j].
lower	a list in the format of v; lower scope of the model.
upper	a list in the format of v; upper scope of the model.
k	penalty, 0 if missing or <0.
direction	forward selection, backward elimination or stepwise.
iter	maximum number of iterations in a numerical process to estimate model parameters.
max.terms	maximum number of terms in the final model.
steps	maximum number of search steps.
tol	convergence tolerance.
ext	whether to perform an extensive search for an "optimal" model with the same number of QTL per phenotype.

Value

a list with the following components:

loglik	log-likelihood of the model
b	estimates of model coefficients
sigma	estimates of residual variance-covariance
v	variables selected in the model

Examples

```
data(etrait)
x<- as.matrix(mdat-1/2)
y<- as.matrix(traits)[,1:3]
v<- list()
upper<- list()
for(k in 1:ncol(y)){
   v[[k]]<- numeric(0)
   upper[[k]]<- 1:ncol(x)
}
## Not run:
o1<- surStep(y, x, v=v, upper=upper, k=19)
o2<- sureStep(o1, y, x, cv=50, ext=FALSE)

# search for optimal model of o1
o3<- sureStep(o1, y, x, cv=Inf, direction="forward", ext=TRUE)

## End(Not run)
```

varGroup *Group variables*

Description

Group variables via variable selection such that the grouped variables are optimal for multivariate analysis.

Usage

```
varGroup(x, z, zscope=NULL, k=qf(0.95,1,nrow(x)-2), kf=k/2,
   method=c("pool","best"), direction=c("both","forward","backward"))
```

Arguments

x	a data matrix/frame. Columns are variables to select from.
z	a data matrix/frame. Columns are variables with elements 1 or not 1 (any others).
zscope	which variables in z to be used for grouping; all if "NULL".

`k`	entry/stay value in backward stepwise.
`kf`	entry/stay value in forward stepwise.
`method`	grouping method at each step: pool all the groups selected from each `zcope` variable, or choose the largest group (see examples).
`direction`	forward selection, backward elimination or both stepwise.

Value

indicators of grouped variables.

Examples

```
data(etrait)
varGroup(traits, mdat, zscope=c(42,55), method="pool")
varGroup(traits, mdat, zscope=c(42,55), method="best")
```

`varSelect` *Variable selection*

Description

Select a specific number of variables via variable selection that are optimal for multivariate analysis.

Usage

```
varSelect(x, group, scope, nv, direction=c("backward","forward"))
```

Arguments

`x`	a data matrix/frame. Columns are variables to select from.
`group`	a grouping indicator of observations.
`scope`	which variables (i.e., columns of `x`) to select from.
`nv`	how many variables to be selected.
`direction`	forward selection or backward elimination.

Value

variables selected in the model.

See Also

`varGroup` and `varStep`

Examples

```
data(etrait)
g55<- varGroup(traits, mdat, zscope=55, method="best")
idx<- sample(1:nrow(traits), replace=FALSE)
varSelect(traits[idx,], group=mdat[,55], scope=1:ncol(traits[idx,]),
   nv=length(g55[[1]]))
```

varStep *Variable selection*

Description

Add a variable, drop a variable, or select a subset of variables via variable selection that are optimal for multivariate analysis.

Usage

```
varAdd1(x, group, vin=NULL, scope=1:ncol(x), k=0)

varDrop1(x, group, vin=1:ncol(x), k=0)

varStep(x, group, scope, k, kf=k/2, direction=c("both",
   "forward","backward"))
```

Arguments

`x`	a data matrix/frame. Columns are variables to select from.
`group`	a grouping indicator of observations.
`vin`	which variables (i.e., columns of `x`) already in model. It defines the initial model.
`scope`	which variables (i.e., columns of `x`) to select from.
`k`	entry/stay value in backward stepwise.
`kf`	entry/stay value in forward stepwise.
`direction`	forward selection, backward elimination or both stepwise.

Value

which variable to add (add1), which variable to drop (drop1), or a subset of variables in the final model (step).

See Also

`varSelect`

Examples

```
data(etrait)
varAdd1(traits, group=mdat[,42], vin=10, scope=1:ncol(traits))
varStep(traits, group=mdat[,42], k=12, scope=1:ncol(traits),
   direction="back")
```

---

varT2 *Hotelling's T^2 test statistic*

Description

Calculate the Hotelling's T^2 test statistic.

Usage

```
varT2(x, group, equalVar=T)
```

Arguments

`x`	a data matrix/frame. Columns are variables to select from.
`group`	a grouping indicator of observations.
`equalVar`	whether assume the same variance-covariance in two groups.

Value

Hotelling's T^2 test statistic.

Examples

```
data(etrait)
varT2(traits, mdat[,42])
```

Index