

# Package ‘mht’

March 23, 2015

**Type** Package

**Title** Multiple Hypothesis Testing for Variable Selection in High-Dimensional Linear Models

**Version** 3.1.2

**Author** Florian Rohart

**Maintainer** Florian Rohart <florian.rohart@gmail.com>

**Description** Multiple Hypothesis Testing For Variable Selection in high dimensional linear models. This package performs variable selection with multiple hypothesis testing, either for ordered variable selection or non-ordered variable selection. In both cases, a sequential procedure is performed. It starts to test the null hypothesis “no variable is relevant”; if this hypothesis is rejected, it then tests “only the first variable is relevant”, and so on until the null hypothesis is accepted.

**License** GPL-3

**Depends** glmnet, Matrix

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-03-23 07:33:03

## R topics documented:

mht-package . . . . .	2
bolasso . . . . .	3
data.scale . . . . .	4
decombaseortho . . . . .	5
dyadiqueordre . . . . .	6
mht . . . . .	8
mht.order . . . . .	10
order.variables . . . . .	12
plot . . . . .	13
plot.bolasso . . . . .	14
predict . . . . .	15
quantilemht . . . . .	16
refit.mht . . . . .	17
refit.mht.order . . . . .	19

---

 mht-package

---

*Multiple hypothesis testing for variable selection*


---

## Description

Multiple hypothesis testing for variable selection in high dimensional linear models.

This package performs variable selection with multiple hypothesis testing, either for ordered variable selection or non-ordered variable selection. In both cases, a sequential procedure is performed. It starts to test the null hypothesis "no variable is relevant"; if this hypothesis is rejected, it then tests "only the first variable is relevant", and so on until the null hypothesis is accepted.

More details are available in the paper 'Multiple hypothesis testing for variable selection', Rohart F. (2011).

## Details

Package:	mht
Type:	Package
Version:	3.1.2
License:	GPL-3
date:	20-03-2015

Two major functions: `mht.order` and `mht` (`proc_ord` and `procbol` in version <3.00, it was changed to give more clarity and flexibility). The first estimates the set of relevant variables for ordered variable selection, e.g. if an apriori of the importance of the variables is known; the last does the same for non-ordered variable selection.

## Author(s)

Florian Rohart  
 Maintainer: [florian.rohart@gmail.com](mailto:florian.rohart@gmail.com)

## References

Multiple hypothesis testing for variable selection; F. Rohart 2011  
 Model-consistent sparse estimation through the bootstrap; F. Bach 2009  
 Adaptive tests of linear hypotheses by model selection; Baraud & al 2002

## Examples

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)
```

```

# mht.order
mod.order=mht.order(x,y,ordre=5:1)
mod.order

# mht
mod=mht(x,y,alpha=c(0.1,0.05),maxordre=15)
mod

## End(Not run)

```

---

bolasso

*Bolasso: Bootstrapped Lasso*


---

## Description

Perform a bootstrapped Lasso on some random subsamplings of the input data

## Usage

```
bolasso(data,Y,mu,m,probbaseuil,penalty.factor,random)
```

## Arguments

data	Input matrix of dimension $n * p$ ; each of the $n$ rows is an observation vector of $p$ variables. The intercept should be included in the first column as $(1, \dots, 1)$ . If not, it is added.
Y	Response variable of length $n$ .
mu	Positive regularization sequence to be used for the Lasso.
m	Number of bootstrap iteration of the Lasso. Default is $m=100$ .
probbaseuil	A frequency threshold for selecting the most stable variables over the $m$ bootstrap iteration of the Lasso. Default is 1.
penalty.factor	Separate penalty factors can be applied to each coefficient. This is a number that multiplies lambda to allow differential shrinkage. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is 1 for all variables except the intercept.
random	optionnal parameter, matrix of size $n*m$ . If <i>random</i> is provided, the $m$ bootstrap samples are constructed from its $m$ columns.

## Details

The Lasso from the [glmnet](#) package is performed with the regularization parameter  $\mu$  over  $m$  bootstrap samples. An appearance frequency is obtained which shows the predictive power of each variable. It is calculated as the number of times a variables has been selected by the Lasso over the  $m$  bootstrap iteration.

**Value**

A 'bolasso' object is returned for which the method plot is available.

data	A list containing: <ul style="list-style-type: none"> <li>• X - The scaled matrix used in the algorithm, the first column being (1,...,1).</li> <li>• Y - the input response vector</li> <li>• means.X - Vector of means of the input data matrix.</li> <li>• sigma.X - Vector of variances of the input data matrix.</li> </ul>
ind	Set of selected variables for the regularization mu and the threshold probaseuil.
frequency	Appearance frequency of each variable; number of times each variables is selected over the m bootstrap iterations.

**References**

Model-consistent sparse estimation through the bootstrap; F. Bach 2009

**See Also**

[plot.bolasso](#), [dyadiqueordre](#)

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(1,5),rep(0,15))
y=x%%beta+rnorm(100)

mod=bolasso(x,y,mu=seq(1.5,0.1,-0.1))
mod

## End(Not run)
```

---

data.scale

*Scale a matrix*

---

**Description**

Scale the data so each column has mean 0 and variance 1. This function is used as a pre-processing step to prep the data for analysis in all functions of the mht package.

**Usage**

```
data.scale(data,warning)
```

**Arguments**

data	Input matrix of dimension $n * p$ ; each row is an observation vector. The intercept should be included in the first column as $(1, \dots, 1)$ . If not, it is added.
warning	Logical value. A warning message is printed if the intercept is added. Default is TRUE.

**Details**

Scale the data so each column has mean 0 and variance 1. If we note  $x$  a column of the output scaled matrix -except the first one which is the intercept, we have  $\sum(x)=0$  and  $\sum(x^2)/n=1$ .

**Value**

data	Scaled data.
intercept	Logical value. TRUE if the intercept was already included in the input data.
means.data	Vector of means of the input data matrix.
sigma.data	Vector of variances of the input data matrix.

**References**

Multiple hypotheses testing for variable selection; F. Rohart 2011

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
res=data.scale(x)
x.scaled=res$data
means.x=res$means.data
sigma.x=res$sigma.data

## End(Not run)
```

---

decompbaseortho	<i>Gram-Schmidt algorithm</i>
-----------------	-------------------------------

---

**Description**

Orthonormalization of an input matrix with the Gram-Schmidt algorithm.

**Usage**

```
decompbaseortho(data)
```

**Arguments**

data                    Input matrix of dimension  $n * p$ ; each column is a variable.

**Details**

Performs an orthonormalization of the input matrix, recording the columns that are linear combination of the previous ones.

**Value**

U                        The orthonormal basis obtained from data.  
 nonind                 Set of variables with no contribution.  
 trueind                Set of variables with contribution.  
 rank                    Rank of the input matrix, calculated with the function [rankMatrix](#) .

**Examples**

```
## Not run:
x1=rnorm(100)
x2=rnorm(100)
x3=cbind(x1,x2,x1+x2)

dec=decompbaseortho(x3)
dec$nonind
dec>trueind
dec$rank

## End(Not run)
```

---

dyadiqueordre                    *Dyadic algorithm to order variables*

---

**Description**

Dyadic algorithm using the Bolasso technique to order the variables

**Usage**

```
dyadiqueordre(data, Y, m, maxordre, var_nselect, showtest, showordre, random)
```

**Arguments**

data                    Input matrix of dimension  $n * p$ ; each of the  $n$  rows is an observation vector of  $p$  variables. The intercept should be included in the first column as  $(1, \dots, 1)$ . If not, it is added.  
 Y                        Response variable of length  $n$ .  
 m                        Number of bootstrap iteration of the Lasso. Default is  $m=100$ .

maxordre	Number of variables to order. Default is $\min(n/2-1, p/2-1)$ .
var_nonselect	Number of variables that don't undergo feature selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
showtest	Logical value. If TRUE, show the number of regularization parameters tested to show the steps of the algorithm. Default is FALSE.
showordre	Logical value. If TRUE, shows the ordered variables at each step of the algorithm. Default is TRUE.
random	optionnal parameter. Matrix of size $n*m$ , the $m$ bootstrap samples are constructed from the $m$ columns.

### Details

The algorithm starts from a large regularization parameter given by one run of Lasso. It proceeds by dyadic splitting until one variable is isolated; e.g one variable alone achieve a frequency of 1; it is the first ordered variable. And so on until maxordre variables are ordered.

### Value

A 'bolasso' object is returned for which the method plot is available.

data	A list containing: <ul style="list-style-type: none"> <li>• X - The scaled matrix used in the algorithm, the first column being <math>(1, \dots, 1)</math>.</li> <li>• Y - the input response vector</li> <li>• means.X - Vector of means of the input data matrix.</li> <li>• sigma.X - Vector of variances of the input data matrix.</li> </ul>
ordre	The order obtained on the variables.
mu	Vector of the positive regularization sequence that was used in the algorithm.
frequency	Matrix of $p$ rows. Appearance frequency of each variable for the regularization parameter in $\mu$ .

### See Also

[bolasso](#), [plot.bolasso](#)

### Examples

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)
mod=dyadiqueordre(x,y,maxordre=15)

## End(Not run)
```

---

mht

---

*Multiple testing procedure for non-ordered variable selection*


---

### Description

Performs multiple hypotheses testing in a linear model

### Usage

```
mht(data,Y,var_nonselect,alpha,sigma,maxordre,ordre,m,show,IT,maxq)
```

### Arguments

data	Input matrix of dimension $n * p$ ; each of the $n$ rows is an observation vector of $p$ variables. The intercept should be included in the first column as $(1, \dots, 1)$ . If not, it is added.
Y	Response variable of length $n$ .
var_nonselect	Number of variables that don't undergo feature selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
alpha	A user supplied type I error sequence. Default is $(0.1, 0.05)$ .
sigma	Value of the variance if it is known; 0 otherwise. Default is 0.
maxordre	Number of variables to be ordered. Default is $\min(n/2-1, p/2-1)$ .
ordre	Several possible algorithms to order the variables, <code>ordre=c("bolasso", "pval", "pval_hd", "FR")</code> . "bolasso" uses the dyadic algorithm with the Bolasso technique <a href="#">dyadiqueordre</a> , "pval" uses the p-values obtained with a regression on the full set of variables (only when $p < n$ ), "pval_hd" uses marginal regression, "FR" uses Forward Regression. Default is "bolasso".
m	Number of bootstrap iteration of the Lasso. Only used if the algorithm is set to "bolasso". Default is $m=100$ .
show	Vector of logical values, <code>show=(showordre, showtest, showresult)</code> . Default is $(1, 0, 1)$ . If <code>showordre==TRUE</code> , show the ordered variables at each step of the algorithm. If <code>showtest==TRUE</code> , show the number of regularization parameters tested to show the advancement of the dyadic algorithm. Only use if the algorithm is set to "bolasso". if <code>showresult==TRUE</code> , show the value of the statistics and the estimated quantile at each step of the procedure.
IT	Number of simulations for the calculation of the quantile. Default is 1000.
maxq	Number of maximum multiple hypotheses testing to perform. Default is $\log(\min(n, p) - 1, 2)$ .



## Details

mht is a two-step procedure that performs variable selection in high dimensional linear model. The first step orders the variables taking into account the vector of observations  $Y$ . The second step finds a cut-off between the relevant variables (high rank) and the irrelevant ones (low rank) through multiple hypotheses testing.

The input `maxordre` is not to be forgotten: the more variables to order, the more difficult for the algorithm to distinguish which noisy variable is more important than another noisy variable. It is advised to limit `maxordre` to  $p/2$  or  $n/2$  if they are large. The parameter `maxq` can be useful for large value of  $n$ , it is advised to limit it to 5-6 in order to minimize computational time (for the calculation of the quantile).

## Value

A 'mht' object is returned for which the methods `refit`, `predict` and `plot` are available.

<code>data</code>	A list containing: <ul style="list-style-type: none"> <li>• <code>X</code> - The scaled matrix used in the algorithm, the first column being <math>(1, \dots, 1)</math>.</li> <li>• <code>Y</code> - the input response vector</li> <li>• <code>means.X</code> - Vector of means of the input data matrix.</li> <li>• <code>sigma.X</code> - Vector of variances of the input data matrix.</li> </ul>
<code>coefficients</code>	Matrix of the estimated coefficients. Each row concerns a specific user level $\alpha$ .
<code>residuals</code>	Matrix of the residuals. Each row concerns a specific user level $\alpha$ .
<code>relevant_var</code>	Set of the relevant variables. Each row concerns a specific user level $\alpha$
<code>fitted.values</code>	Matrix of the fitted values, each column concerns a specific user level $\alpha$ .
<code>ordre</code>	Order obtained on the <code>maxordre</code> variables.
<code>ordrebeta</code>	The full order on all the variables.
<code>kchap</code>	Vector containing the length of the estimated set of relevant variables, for each values of $\alpha$ .
<code>quantile</code>	The estimated quantiles used in the second step of the procedure.
<code>call</code>	The call that produced this object.

## References

Multiple hypotheses testing for variable selection; F. Rohart 2011

## See Also

[predict.mht](#), [refit.mht](#), [plot.mht](#)

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)

mod=mht(x,y,alpha=c(0.1,0.05),maxordre=15)
mod

## End(Not run)
```

---

mht.order

---

*Multiple testing procedure for ordered variable selection*


---

**Description**

Performs multiple hypotheses testing for ordered variable selection.

**Usage**

```
mht.order(data,Y,ordre,var_nonselect,alpha,IT,sigma,showresult)
```

**Arguments**

data	Input matrix of dimension $n * p$ ; each of the $n$ rows is an observation vector of $p$ variables. The intercept should be included in the first column as $(1, \dots, 1)$ . If not, it is added.
Y	Response variable of length $n$ .
ordre	Vector from which the variables are to be ordered, it can be a partial order. If absent, data is considered to be already ordered; Default is $(1, 2, \dots, p)$ .
var_nonselect	Number of variables that don't undergo feature selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
alpha	A user supplied type I error sequence. Default is $\alpha=(0.1, 0.05)$
IT	Number of simulations in the calculation of the quantile. Default is 10000.
sigma	Value of the variance if it is known; 0 otherwise. Default is 0.
showresult	Logical value. if TRUE, shows the value of the statistics and the estimated quantile at each step of the procedure. Default is TRUE.

**Details**

The details of the procedure are in 'Multiple hypotheses testing for variable selection; F. Rohart 2011'. If `showresult=TRUE`, the statistics and quantile are printed through the algorithm. If the statistic is greater than the quantile, the test is rejected (takes the value 1). The procedure stops when the null hypothesis is accepted (all alternative hypotheses are 0).

The statistics to test the null hypotheses are different whether the variance `sigma` is known.

**Value**

A 'mht.order' object is returned for which the methods `predict`, `refit` and `plot` are available.

<code>data</code>	A list containing: <ul style="list-style-type: none"> <li>• <code>X</code> - The scaled matrix used in the algorithm, the first column being (1,...,1).</li> <li>• <code>Y</code> - the input response vector</li> <li>• <code>means.X</code> - Vector of means of the input data matrix.</li> <li>• <code>sigma.X</code> - Vector of variances of the input data matrix.</li> </ul>
<code>coefficients</code>	Matrix of the estimated coefficients. Each row concerns a specific user level alpha.
<code>residuals</code>	Matrix of the residuals. Each row concerns a specific user level alpha.
<code>relevant_var</code>	Set of the relevant variables. Each row concerns a specific user level alpha
<code>fitted.values</code>	Matrix of the fitted values, each column concerns a specific user level alpha.
<code>kchap</code>	Vector containing the length of the estimated set of relevant variables, for each values of alpha.
<code>quantile</code>	The estimated type I error to be used in the second step of the procedure in order to have a test of level alpha, each column stands for one test. See F.Rohart (2011) for details.
<code>call</code>	The call that has been used.

**References**

Adaptive tests of linear hypotheses by model selection; Baraud & al 2002  
 Multiple hypotheses testing for variable selection; F. Rohart 2011

**See Also**

[predict.mht.order](#), [refit.mht.order](#), [plot.mht.order](#)

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)

mod.order=mht.order(x,y,ordre=5:1,alpha=c(0.1,0.05))
mod.order

## End(Not run)
```

---

order.variables	<i>Rank the variables</i>
-----------------	---------------------------

---

### Description

Gives an order to the variables and rearrange the input matrix following that order.

### Usage

```
order.variables(data,Y,maxordre,ordre=c("bolasso","pval","pval_hd","FR"),
var_nonselect,m,showordre)
```

### Arguments

data	Input matrix of dimension $n * p$ ; each of the $n$ rows is an observation vector of $p$ variables. The intercept should be included in the first column as $(1, \dots, 1)$ . If not, it is added.
Y	Response variable of length $n$ .
maxordre	Number of variables to be ordered. Default is $\min(n/2-1, p/2-1)$ .
ordre	Several possible algorithms to order the variables, <code>ordre=c("bolasso","pval","pval_hd","FR")</code> . "bolasso" uses the dyadic algorithm with the Bolasso technique <a href="#">dyadiqueordre</a> , "pval" uses the p-values obtained with a regression on the full set of variables (only when $p < n$ ), "pval_hd" uses marginal regression, "FR" uses Forward Regression. Default is "bolasso".
var_nonselect	Number of variables that don't undergo feature selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
m	Number of bootstrapped iteration of the Lasso. Only use if the algorithm is set to "bolasso". Default is $m=100$ .
showordre	If <code>showordre=TRUE</code> , show the variables being ordered at each step of the algorithm.

### Details

Rank the variables of data taking into account the vector of observations  $Y$  and rearrange the input matrix following that order.

### Value

data	A list containing: <ul style="list-style-type: none"> <li>• X - The scaled matrix used in the algorithm, the first column being <math>(1, \dots, 1)</math>.</li> <li>• Y - the input response vector</li> <li>• means.X - Vector of means of the input data matrix.</li> <li>• sigma.X - Vector of variances of the input data matrix.</li> </ul>
data_ord	Input data matrix rearranged by ORDREBETA

ORDRE	Gives the maxordre most important variables of the data matrix.
ORDREBETA	Gives the order on all the variables of the data matrix (either arbitrary completion of ORDRE - 'Bolasso' and 'FR', or the true order - 'pval' and 'pval_hd').

## References

Multiple hypotheses testing for variable selection; F. Rohart 2011

## Examples

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)

res.bolasso=order.variables(x,y,maxordre=15,ordre="bolasso")
res.pval=order.variables(x,y,ordre="pval")
res.FR=order.variables(x,y,maxordre=15,ordre="FR")
res.pval.hd=order.variables(x,y,maxordre=15,ordre="pval_hd")

## End(Not run)
```

---

plot *graphical output for a mht or mht.order object*

---

## Description

Graphical output for a mht or mht.order object. Four plots (selectable by which.plot) are currently available: a plot of the fitted values against the true values, a plot of the residuals against the fitted values, a Normal Q-Q plot, and a barplot showing the coefficients.

## Usage

```
## S3 method for class 'mht'
plot(x,which.plot=1:4,...)
## S3 method for class 'mht.order'
plot(x,which.plot=1:4,...)
```

## Arguments

x	Object of class "mht" as obtained from <code>mht</code> or "mht.order" as obtained from <code>mht.order</code> .
which.plot	if a subset of the plots is required, specify a subset of the numbers 1:4.
...	not used.

**Details**

The four plots are shown for each level alpha of the mht or mht.order object.

For the Normal Q-Q plot, the standardized residuals are used. They are calculated as  $R[i]/\sqrt{\text{var}(R)}$ .

**See Also**

[mht](#), [mht.order](#)

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)

# mht
mod=mht(x,y,alpha=c(0.1,0.05),maxordre=15)
plot(mod)

# mht.order
mod.order=mht.order(x,y,ordre=5:1,alpha=c(0.1,0.05))
plot(mod.order)

## End(Not run)
```

---

plot.bolasso

*graphical output for a bolasso object*

---

**Description**

graphical output for a bolasso object. Plot the frequency of selection of each variable depending on the regularization parameter mu from the "bolasso" object.

**Usage**

```
## S3 method for class 'bolasso'
plot(x,...)
```

**Arguments**

x                    Object of class "bolasso". As obtained from [bolasso](#).  
 ...                   not used.

**Details**

Plot the frequency of selection of all variables depending on the regularization parameter.

**See Also**

[bolasso](#), [dyadiqueordre](#)

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)

# BoLasso
mod=bolasso(x,y,mu=seq(1.5,0.1,-0.1))
plot(mod)

## End(Not run)
```

---

predict	<i>Predict a mht or mht.order object</i>
---------	--

---

**Description**

Predict a mht or mht.order object for new data newx

**Usage**

```
## S3 method for class 'mht'
predict(object,newx,level,...)
## S3 method for class 'mht.order'
predict(object,newx,level,...)
```

**Arguments**

object	Object of class "mht.order" as obtained from <a href="#">mht.order</a> or "mht" as obtained from <a href="#">mht</a> .
newx	Data matrix of size n*p.
level	Level of the prediction interval. Default is 0.95.
...	not used.

**Details**

The prediction is available for each level alpha of the object. The prediction values and the prediction interval are derived from the [predict.lm](#) function. If newx is missing, the fitted values of the object are returned.

**Value**

Array of predicted values and prediction interval. The third dimension is relative to the type I error alpha -from the initial object-. For each alpha, a matrix with column names fit, lwr, and upr (from [predict.lm](#))

**See Also**

[mht](#), [mht.order](#), [predict.lm](#)

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)

# mht
mod=mht(x,y,alpha=c(0.1,0.05),maxordre=15)

# predict without new data: gives the fitted values
pred=predict(mod)

# predict with new data
pred=predict(mod,newx=matrix(rnorm(40*20),40,20))
pred

## End(Not run)
```

---

quantilemht

*Calculation of the quantiles for the mht procedure*

---

**Description**

Calculation of the quantiles for the mht procedure

**Usage**

```
quantilemht(data,k,alpha,IT,maxq,sigma)
```

**Arguments**

data	Input matrix of dimension $n \times p$ ; each row is an observation vector.
k	Positive integer.
alpha	A user supplied type I error sequence. Default is (0.1,0.05).
IT	Number of simulations in the calculation of the quantile. Default is 1000.
maxq	Number of maximum multiple hypotheses testing to do. Default is $\log(\min(n,p)-1,2)$ .
sigma	Value of the variance if it is known; 0 otherwise. Default is 0.



**Details**

Calculation of the quantile of the statistic that tests the null hypothesis  $H_k: E(Y) = X_1\beta_1 + X_2\beta_2 + \dots + X_k\beta_k$ , i.e there is no more signal remaining in the variables (k+1,...,p).

**Value**

quantile	The estimated quantiles.
nbrprob	Number of times there was not enough simulation to estimate the alpha-quantile.

---

refit.mht	<i>Refit a mht object</i>
-----------	---------------------------

---

**Description**

Refit a mht object for a new observation Ynew

**Usage**

```
## S3 method for class 'mht'
refit(object, Ynew, var_nonselect, sigma, maxordre, ordre, m, show, IT, ...)
```

**Arguments**

object	Object of class "mht".
Ynew	Response variable of length n.
var_nonselect	Number of variables that don't undergo feature selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
sigma	Value of the variance if it is known; 0 otherwise. Default is 0.
maxordre	Number of variables to be ordered. Default is $\min(n/2-1, p/2-1)$ .
ordre	Several possible algorithms to order the variables, <code>ordre=c("bolasso", "pval", "pval_hd", "FR")</code> . "bolasso" uses the dyadic algorithm with the Bolasso technique <a href="#">dyadiqueordre</a> , "pval" uses the p-values obtained with a regression on the full set of variables (only when $p < n$ ), "pval_hd" uses marginal regression, "FR" uses Forward Regression. Default is "bolasso".
m	Number of bootstrapped iteration of the Lasso. Only use if the algorithm is set to "bolasso". Default is $m=100$ .
show	Vector of logical values, <code>show=(showordre, showtest, showresult)</code> . Default is (1,0,1). If <code>showordre=TRUE</code> , shows the variables being ordered at each step of the algorithm. If <code>showtest=TRUE</code> , the number of regularization parameters tested is printed to show the advancement of the dyadic algorithm. Only used if the algorithm is set to "bolasso". if <code>showresult=TRUE</code> , shows the value of the statistics and the estimated quantile at each step of the procedure.
IT	Number of simulations for the calculation of the quantile. Default is 1000.
...	not used.

**Details**

maxq is not a parameter of refit.mht as the same number of alternative is necessary for a refit of the model.

For more details, see [mht](#).

**Value**

A 'mht' object is returned.

data	A list containing: <ul style="list-style-type: none"> <li>• X - The scaled matrix used in the algorithm, the first column being (1,...,1).</li> <li>• Y - the input response vector</li> <li>• means.X - Vector of means of the input data matrix.</li> <li>• sigma.X - Vector of variances of the input data matrix.</li> </ul>
coefficients	Matrix of the estimated coefficients. Each row concerns a specific user level alpha.
residuals	Matrix of the residuals. Each row concerns a specific user level alpha.
relevant_var	Set of the relevant variables for each alpha.
fitted.values	Matrix of the fitted values, each column concerns a specific user level alpha.
ordre	Order obtained on the maxordre variable.
ordrebeta	The full order on all the p variables.
kchap	Vector containing the length of the estimated set of relevant variables, for the matrix containing the intercept, for each values of alpha.
quantile	The estimated quantiles used in the second step of the procedure.
call	The call that has been used.
call.old	The call that produced the initial 'object'.

**Examples**

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)
ynew=x%%beta+rnorm(100)

# mht
mod=mht(x,y,alpha=c(0.1,0.05),maxordre=15)

# refit mht on a new vector of observation
mod2=refit(mod,ynew,maxordre=15)

## End(Not run)
```

---

refit.mht.order	<i>Refit a mht.order object</i>
-----------------	---------------------------------

---

**Description**

Refit a `mht.order` object for a new observation `Ynew` and/or a new order `ordrenew`

**Usage**

```
## S3 method for class 'mht.order'
refit(object, Ynew, ordrenew, IT, var_nonselect, sigma, showresult, ...)
```

**Arguments**

<code>object</code>	Object of class "mht.order" as obtained from <a href="#">mht</a> .
<code>Ynew</code>	Response variable of length <code>n</code> .
<code>ordrenew</code>	Vector from which the variables are to be ordered, it can be a partial order. If absent, data is considered to be already ordered; Default is $(1, 2, \dots, p)$ .
<code>IT</code>	Number of simulations in the calculation of the quantile. Default is 10000.
<code>var_nonselect</code>	Number of variables that don't undergo feature selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
<code>sigma</code>	Value of the variance if it is known; 0 otherwise. Default is 0.
<code>showresult</code>	Logical value. if TRUE, show the value of the statistics and the estimated quantile at each step of the procedure. Default is TRUE.
<code>...</code>	not used.

**Details**

see [mht.order](#) for details.

**Value**

A 'mht.order' object is returned.

<code>data</code>	A list containing: <ul style="list-style-type: none"> <li>• <code>X</code> - The scaled matrix used in the algorithm, the first column being <math>(1, \dots, 1)</math>.</li> <li>• <code>Y</code> - the input response vector</li> <li>• <code>means.X</code> - Vector of means of the input data matrix.</li> <li>• <code>sigma.X</code> - Vector of variances of the input data matrix.</li> </ul>
<code>coefficients</code>	Matrix of the estimated coefficients. Each row concerns a specific user level <code>alpha</code> .
<code>residuals</code>	Matrix of the residuals. Each row concerns a specific user level <code>alpha</code> .
<code>relevant_var</code>	Set of the relevant variables. Each row concerns a specific user level <code>alpha</code>

<code>fitted.values</code>	Matrix of the fitted values, each column concerns a specific user level alpha.
<code>kchap</code>	Vector containing the length of the estimated set of relevant variables, for each values of alpha.
<code>quantile</code>	The estimated type I error to be used in the second step of the procedure in order to have a test of level alpha, each column stands for one test. See F.Rohart (2011) for details.
<code>call</code>	The call that has been used.
<code>call.old</code>	The call that produced the initial 'object'.

### Examples

```
## Not run:
x=matrix(rnorm(100*20),100,20)
beta=c(rep(2,5),rep(0,15))
y=x%%beta+rnorm(100)
ynew=x%%beta+rnorm(100)

# mht.order
mod.order=mht.order(x,y)

# refit mht.order on a new vector of observation
mod.refit=refit(mod.order,ynew)

## End(Not run)
```

# Index

`bolasso`, [3](#), [7](#), [14](#), [15](#)

`data.scale`, [4](#)

`decompbaseortho`, [5](#)

`dyadiqueordre`, [4](#), [6](#), [8](#), [12](#), [15](#), [17](#)

`glmnet`, [3](#)

`mht`, [2](#), [8](#), [13–16](#), [18](#), [19](#)

`mht-package`, [2](#)

`mht.order`, [2](#), [10](#), [13–16](#), [19](#)

`order.variables`, [12](#)

`plot`, [13](#)

`plot.bolasso`, [4](#), [7](#), [14](#)

`plot.mht`, [9](#)

`plot.mht.order`, [11](#)

`predict`, [15](#)

`predict.lm`, [15](#), [16](#)

`predict.mht`, [9](#)

`predict.mht.order`, [11](#)

`quantilemht`, [16](#)

`rankMatrix`, [6](#)

`refit(refit.mht)`, [17](#)

`refit.mht`, [9](#), [17](#)

`refit.mht.order`, [11](#), [19](#)