Package 'PSEA'

October 9, 2015

Version 1.2.0

Date 2014-04-21

Title Population-Specific Expression Analysis.

Author Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

Maintainer Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

Imports Biobase, MASS

Suggests BiocStyle

Description Deconvolution of gene expression data by
Population-Specific Expression Analysis (PSEA).

biocViews Software

License Artistic-2.0

ZipData no

NeedsCompilation no

R topics documented:

| coefmat | *Extracts fitted coefficients.* |

Description

Takes a list of fitted models (lm objects) and returns coefficients in matrix format.

Usage

```
coefmat(lst,regressors)
```

Arguments

lst	list of lm objects.
regressors	character vector. Names of the coefficients to extract.

Details

Simple wrapper function that returns fitted coefficients.

The names of the coefficient to extract are matched in names(lst[[i]]$coef).

The column in the matrix of extracted coefficients are named by prepending "coef." to the regressor names.

Value

coefm	numeric matrix. Matrix of extracted coefficients with one row for each model in the list and one column for each targeted coefficient.

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

See Also

pvalmat.

Examples

```
## Load example expression data (variable "expression")
## and phenotype data (variable "groups")
data("example")

## Four cell population-specific reference signals
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
                "201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)

astro_probesets <- list("203540_at",c("210068_s_at","210906_x_at"),
```

```
                 "201667_at")
astro_reference <- marker(expression, astro_probesets)

oligo_probesets <- list(c("211836_s_at","214650_x_at"),"216617_s_at",
                 "207659_s_at",c("207323_s_at","209072_at"))
oligo_reference <- marker(expression, oligo_probesets)

micro_probesets <- list("204192_at", "203416_at")
micro_reference <- marker(expression, micro_probesets)

## Stepwise model selection for 2 transcripts (202429_s_at and
## 200850_s_at) and focusing on control samples (i.e. groups == 0)
lmlist <- swlm(expression[c("202429_s_at", "200850_s_at"),],
subset = which(groups == 0),
upper = formula(~neuron_reference + astro_reference +
oligo_reference + micro_reference))

coefmat(lmlist, c("(Intercept)", "neuron_reference", "astro_reference",
"oligo_reference", "micro_reference"))
```

---

crplot	*Component-plus-residual (CR) plot*

Description

Component-plus-residual (CR) plot for quantitative variables and group-specific effects.

Usage

```
crplot(lm,quantv,g=NULL,newplot=TRUE,col=c(1,2,4),xlab=NULL,ylab='CR',...)
```

Arguments

lm	object of class "lm". Fitted model.
quantv	character. Name of the quantitative independent variable ("component" to be plotted).
g	character vector. Interaction regressors coding for group-specific effects in the model.
newplot	logical. If TRUE (default) a new plot (graphics device) is opened.
col	integer vector. Colors of groups. Defaults to 1 (black), 2 (red) and 4 (blue) for group 1, 2 and 3, respectively.
xlab	character. x-axis label. Absent by default.
ylab	character. y-axis label. "CR" by default.
...	Further arguments passed to plot()

Details

In the fitted model (lm), group-specific effects are specified by interaction regressors. In this case, an interaction regressor is a quantitative variable multiplied by a binary variable (i.e. where 0s represent samples of group 1 and 1s represent samples of group 2).

Predicted components are not centered and the intercept of the CR is 0 by definition. This function thus differs from cr.plot in the car package.

Value

NULL   This function is used for plotting.

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

References

Kuhn A, Thu D, Waldvogel HJ, Faull RL, Luthi-Carter R. Population-specific expression analysis (PSEA) reveals molecular changes in diseased brain. Nat Methods 2011, 8(11):945-7

Examples

```
## Load example expression data (variable "expression")
## for 23 transcripts and 41 samples, and associated
## phenotype (i.e. group) information (variable "groups")
data("example")

## The group data is encoded as a binary vector where
## 0s represent control samples (first 29 samples) and
## 1s represent disease samples (last 12 samples)
groups

## Neuronal reference signals (i.e. quantitative variable)
## and group-specific change in neuronal expression
## (i.e. interaction regressor)
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
                "201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)
neuron_difference <- groups * neuron_reference

## Fit an expression model containing neuronal expression and
## neuron-specific change in expression between control and
## disease samples
model <- lm(expression["202429_s_at",] ~ neuron_reference +
neuron_difference)

## Visualize the dependence on the neuronal reference signal and
## the group-specific effect (decreased neuronal expression in
## disease samples). Black and red dots represent control and
## disease samples, respectively.
crplot(model, "neuron_reference", g="neuron_difference")
```

---

em_quantvg | *Specifies particular models useful for gene expression modeling*

Description

Enumerates sets of integer that specify variables to include in each model of the family.

Usage

```
em_quantvg(vi,tnv=NULL,ng=1,sk=NULL,mnr=NULL,int=TRUE)
```

Arguments

vi
: integer vector. Indices of the quantitative variables in the full model matrix.

tnv
: integer. Total number of quantitative variables (excluding the intercept) in the full model matrix (if (int=TRUE) tnv>=(max(vi)-1). If (int=FALSE) tnv>=max(vi)). By default: tnv=(max(vi)-1) if (int=TRUE) and tnv=max(vi) if (int=FALSE).

ng
: integer. Number of samples groups: 1 (default), 2 or 3.

sk
: character. Indicates if the models with group-specific regressors defining group 2 ("skip2") should be eliminated or if models containing group-specific regressors for groups 2 and 3 (simultaneously) should be prohibited ("skip23"). These apply only if ng=3.

mnr
: integer. Maximal total number of regressors (including the intercept if "int" is TRUE) in the enumerated models.

int
: logical. If TRUE (default) the first column of the full model matrix is assumed to be an intercept.

Details

We specify a family of linear models corresponding to a (multiple) regression on selected quantitative variables and up to 3 different treatment groups. The function enumerates all subset models with up to 1 (i.e. 0 or 1) group-specific change in any of selected the quantitative variables. Regressors coding for a group-specific change in a given quantitative variable are only allowed if the correponding quantitative variable is present in the model.

The function returns sets of integers that define models by specifying indices to the columns of a full model matrix. The full model matrix can contain more quantitative variables than those selected for building the enumeration.

The full model matrix (with columns as regressors) is assumed to contain all quantitative variables first, followed by the interaction regressors coding for differences between sample group 1 and 2, possibly followed by the interaction regressors coding for differences between group 1 and 3. The maximal number of sample groups is 3. The order of interaction regressors for a given group is assumed to be the same as for the quantitative variables.

The first column of the full model matrix is assumed to be an intercept. If the full model matrix does not contain an intercept, the argument "int" should be set to FALSE (and the returned models will not include an intercept).

Value

mi   list of integer vectors. Each vector defines a model by specifying column indices
                         in the full model matrix.

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

References

Kuhn A, Thu D, Waldvogel HJ, Faull RL, Luthi-Carter R. Population-specific expression analysis
(PSEA) reveals molecular changes in diseased brain. Nat Methods 2011, 8(11):945-7

See Also

[fmm](#).

Examples

```
## Load example expression data (variable "expression")
## for 23 transcripts and 41 samples, and associated
## phenotype (i.e. group) information (variable "groups")
data("example")

## The group data is encoded as a binary vector where
## 0s represent control samples (first 29 samples) and
## 1s represent disease samples (last 12 samples)
groups

## Four cell population-specific reference signals
## (i.e. quantitative variable)
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
                "201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)

astro_probesets <- list("203540_at",c("210068_s_at","210906_x_at"),
                "201667_at")
astro_reference <- marker(expression, astro_probesets)

oligo_probesets <- list(c("211836_s_at","214650_x_at"),"216617_s_at",
                "207659_s_at",c("207323_s_at","209072_at"))
oligo_reference <- marker(expression, oligo_probesets)

micro_probesets <- list("204192_at", "203416_at")
micro_reference <- marker(expression, micro_probesets)

## Full model matrix with an intercept, 4 quantitative variables and
## group-specific (disease vs control) differences for the
## 4 quantitative variables
model_matrix <- fmm(cbind(neuron_reference,astro_reference,
oligo_reference, micro_reference), groups)
```

```
## Enumerate all possible models with any subset of the 4 reference signals
## (quantitiatve variables) and at most 1 group-specific effect
## (interaction regressor)
model_subset <- em_quantvg(c(2,3,4,5), tnv=4, ng=2)

## There are 48 models
length(model_subset)

## For instance the 17th model in the list contains an intercept (column 1 in model_matrix), the neuronal referenc
model_subset[[17]]
```

---

expression	*Expression data.*

Description

Numeric matrix containing expression data for 23 transcripts (rows) and 41 samples (columns).

The data correspond to samples 1, 101, 11, 126, 14, 15, 17, 2, 20, 21, 52, 64, 8, H104, H109, H113, H115, H117, H118, H120, H121, H124, H126, H128, H129, H131, H132, H137, H85, 12, 13, 3, 7, HC103, HC105, HC53, HC55, HC74, HC81, HC83, HC86 for which expression data has been deposited as GSE3790 at Gene Expression Omnibus http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3790. Column names in the expression data matrix contain the corresponding CEL file names.

The first 29 samples are control samples and the next 12 samples are disease samples.

Format

Numeric matrix with 23 rows and 41 columns.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3790

---

fmm	*Full model matrix*

Description

Generates a model matrix containing (quantitative) independent variables and interaction regressors coding for group-specific effects for each of tthe quantitative variable.

Usage

```
fmm(v,d,int=TRUE)
```

Arguments

v	n-by-m numeric matrix. Matrix containing m quantitative variables as columns (i.e. n is the number of samples).
d	n-by-o numeric matrix. Matrix containing o binary regressors (encoding o group-specific differences) as columns. n is the number of samples.
int	logical. If TRUE (default) an intercept is included as the first column of the full model matrix.

Details

Quantitative variables are specified via the argument "v". Sample groups are specified using binary (i.e. 0,1) variables (argument "d"). The returned model matrix represents a linear model with "m" quantitative variables and "o" group effects (corresponding to "o"+1 sample groups).

Specifically, the columns of the full model matrix contain (in this order): an intercept (if "int" is TRUE), the quantitative variables in the same order as they are provided in the input, group-specific effects for each quantitative variable (encoded as interaction regressors and in the same order as the quantitative variable are provided in the input).

An interaction regressor corresponds a quantitative variable multiplied by a (group-specific) binary variable. It encodes the difference (for the particular variable) between a specific group and the reference group.

Value

fmm	numeric matrix. The full model matrix. Its dimension is is n-by-(m*(o+1)+1) if int is TRUE and n-by-(m*(o+1)) if int is FALSE.

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

Examples

```
## Load example expression data (variable "expression")
## for 23 transcripts and 41 samples, and associated
## phenotype (i.e. group) information (variable "groups")
data("example")

## The group data is encoded as a binary vector where
## 0s represent control samples (first 29 samples) and
## 1s represent disease samples (last 12 samples)
groups

## Four cell population-specific reference signals
## (i.e. quantitative variable)
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
                "201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)

astro_probesets <- list("203540_at",c("210068_s_at","210906_x_at"),
```

```
                "201667_at")
astro_reference <- marker(expression, astro_probesets)

oligo_probesets <- list(c("211836_s_at","214650_x_at"),"216617_s_at",
                "207659_s_at",c("207323_s_at","209072_at"))
oligo_reference <- marker(expression, oligo_probesets)

micro_probesets <- list("204192_at", "203416_at")
micro_reference <- marker(expression, micro_probesets)

## Full model matrix with an intercept, 4 quantitative variables and
## group-specific (disease vs control) differences for the
## 4 quantitative variables
fmm(cbind(neuron_reference, astro_reference, oligo_reference,
micro_reference), groups)
```

groups	*Binary vector encoding sample group information.*

Description

Numeric vector of length 41 containing 0s and 1s. 0s and 1s represent samples belonging to sample group 1 and 2, respectively.

This vector actually correspond to samples 1, 101, 11, 126, 14, 15, 17, 2, 20, 21, 52, 64, 8, H104, H109, H113, H115, H117, H118, H120, H121, H124, H126, H128, H129, H131, H132, H137, H85, 12, 13, 3, 7, HC103, HC105, HC53, HC55, HC74, HC81, HC83, HC86 for which expression data has been deposited as GSE3790 at Gene Expression Omnibus http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3790. 0s are control samples and 1s are disease samples.

Format

Numeric vector of length 41.

Source

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3790

lmfitst	*Fits a set of models and performs model selection.*

Description

For each response (dependent variable), the function fits a set of models and identify the best models based on AIC.

Usage

```
lmfitst(y,fmdlm,st,subset=NULL,d=2,lm=TRUE)
```

Arguments

y	numeric matrix or ExpressionSet. Matrix with response (dependent) variable as columns. If an ExpressionSet is provided the response matrix is extracted with the function exprs and transposed.
fmdlm	numeric matrix. Full model matrix with regressors as columns.
st	list of integer vectors. Each vector defines a model by specifying column indices in fmdlm.
subset	Integer vector. Subset of samples to be used for fitting. By default all samples are used.
d	numeric. Maximal distance (in AIC units) to the model with smallest AIC (defaults to 2). Models with an AIC difference less than or equal to d are returned.
lm	logical. If TRUE (default) the function also returns the fit (as returned by the function lm) obtained for the best model obtained for each response. If FALSE, the function only returns the indices (in st) corresponding to the selected models.

Details

This function is useful for performing model selection. It fits all possible models of a given set (instead of heuristically searching for the best one for instance) and compares them using Akaike Information Criterion (AIC).

For each response, all specified models are tested and all models obtaining an AIC within "d" AIC units of the model with smallest AIC are identified. These models are ordered by increasing number of regressors. Models with the same number of regressors are ordered by increasing AIC. The first model in this ordered list is considered the best model. Optionally the full fit (as returned by the function lm) of the best model is returned.

The formula for AIC follows the implementation of the function extractAIC and is: $n * \log(RSS / n) + 2 * edf$, where n is the number of observations, RSS is the residual sum of squares and edf the number of free parameters.

Model fitting is implemented with lm.fit and NAs are not allowable in the response matrix y.

Value

wcrto	list of integer vector (of length equal to the number of responses). Each integer vector represents the models (identified as indices into st) within d AIC units of the model with smallest AIC.
ft	list of lm objects. Each lm object contains the fit for the best model (defined as the model with the smallest number of regressors within d AIC units of the model with smallest AIC). Only returned if lm is TRUE (default).

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

References

Kuhn A, Thu D, Waldvogel HJ, Faull RL, Luthi-Carter R. Population-specific expression analysis (PSEA) reveals molecular changes in diseased brain. Nat Methods 2011, 8(11):945-7

See Also

marker,swlm.

Examples

```
## Load example expression data (variable "expression")
## for 23 transcripts and 41 samples, and associated
## phenotype (i.e. group) information (variable "groups")
data("example")

## The group data is encoded as a binary vector where
## 0s represent control samples (first 29 samples) and
## 1s represent disease samples (last 12 samples)
groups

## Four cell population-specific reference signals
## (i.e. quantitative variable)
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
                "201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)

astro_probesets <- list("203540_at",c("210068_s_at","210906_x_at"),
                "201667_at")
astro_reference <- marker(expression, astro_probesets)

oligo_probesets <- list(c("211836_s_at","214650_x_at"),"216617_s_at",
                "207659_s_at",c("207323_s_at","209072_at"))
oligo_reference <- marker(expression, oligo_probesets)

micro_probesets <- list("204192_at", "203416_at")
micro_reference <- marker(expression, micro_probesets)

## Full model matrix with an intercept, 4 quantitative variables and
## group-specific (disease vs control) differences for the
## 4 quantitative variables
model_matrix <- fmm(cbind(neuron_reference, astro_reference,
oligo_reference, micro_reference), groups)

## Enumerate all possible models with any subset of the 4 reference signals
## (quantitiatve variables) and at most 1 group-specific effect
## (interaction regressor)
model_subset <- em_quantvg(c(2,3,4,5), tnv=4, ng=2)

## AIC-based model selection for 2 transcripts of interest
## (202429_s_at and 200850_s_at). For the first one, the selected
## model contains an intercept, the neuronal reference signal and
## a neuron-specific change (i.e. model 17 in model_subset
```

```
## corresponding to columns 1, 2 and 6 in model_matrix). For the
## second transcript, the selected model contains an intercept and
## the astrocytic reference signal (i.e. model 3 in model_subset
## corresponding to columns 1 and 3 in model_matrix)
lmfitst(t(expression[c("202429_s_at", "200850_s_at"),]),
model_matrix, model_subset)
```

---

marker   *Reference signal*

Description

Generates a population-specific reference signal.

Usage

```
marker(expr,id,sampleSubset=NULL,targetMean=1)
```

Arguments

expr   Numeric matrix or ExpressionSet. The matrix contains the expression data with
              transcripts (i.e. for a microarray, probes) as rows and samples as columns. Row
              names should be defined and are used to determine the transcripts used to gen-
              erate the reference signal (specified with the argument id). If an ExpressionSet
              is provided the expression data is extracted with the function exprs.

id   List of strings. Names of the transcripts to use to generate the reference sig-
              nal. Names correspond to row names in expr if it is a matrix or row names in
              exprs(expr) if it is an ExpressionSet).

sampleSubset Integer vector. Represents a subset of samples (specified as column indices
              in expr) to use to normalize the reference signal (see Details). By default all
              samples are used.

targetMean   Numeric. Mean of the reference signal (defaults to 1).

Details

A reference signal for a given population is generated by averaging the expression of multiple
transcripts that are expressed in a cell population-specific manner.

The reference signal is scaled to have an average value determined by targetMean (1 by default).
The argument sampleSubset restricts the samples to take into account for normalization (i.e. the
average of the reference signal over sampleSubset will be the value specified by targetMean, but the
reference signal will still be of length equal to the number of columns in expr).

The expression value of each transcript is normalized before averaging over multiple transcripts so
that they have same weight in the final reference signal (i.e to avoid that highly expressed transcripts
dominate the reference signal).

The names of transcripts to be used are specified as items of a list (id). If the expression matrix
contains multiple measures of the same transcript (for instance several probes of a microarray mea-
suring the same transcript) they can be averaged before taking the average over different transcripts.

Thereby every transcript has the same weight in the final reference signal, independently of the number of probes used to measure it. Multiple measures of the same transcript are specified as a vector in the list.

Value

refSignal   Numeric vector of the same length as the number of columns in expr (or in exprs(expr) if expr is provided as an ExpressionSet). Reference signal (see Details).

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

References

Kuhn A, Thu D, Waldvogel HJ, Faull RL, Luthi-Carter R. Population-specific expression analysis (PSEA) reveals molecular changes in diseased brain. Nat Methods 2011, 8(11):945-7

Examples

```
## Load example expression data
data("example")

## List of probes corresponding to neuronal genes:
## 221805_at, 221801_x_at, 221916_at all measure NEFL expression.
## 201313_at, 210040_at, 205737_at, 210432_s_at measure
## ENO2, SLC12A5, KCNQ2 and SCN3A expression, respectively.
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
"201313_at", "210040_at", "205737_at", "210432_s_at")

## Neuronal reference signal
neuron_reference <- marker(expression, neuron_probesets)
```

---

pvalmat   *Extracts p-values*

Description

Takes a list of fitted models (lm objects) and obtains the p-values associated with each fit.

Usage

```
pvalmat(lst,regressors)
```

Arguments

lst   list of lm objects.

regressors   character vector. Names of the coefficients for which p-values are extracted.

Details

Simple wrapper function that returns p-values extracted from summary(lm).

The coefficient names corresponding to the p-values to be extracted are matched based on their names in rownames(summary(lst[[i]])$coef).

Value

pvalm   numeric matrix. Matrix of extracted p-values with one row for each model in the list and one column for each targeted coefficient.

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

See Also

coefmat.

Examples

```
## Load example expression data (variable "expression")
## and phenotype data (variable "groups")
data("example")

## Four cell population-specific reference signals
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
"201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)

astro_probesets <- list("203540_at",c("210068_s_at","210906_x_at"),
"201667_at")
astro_reference <- marker(expression, astro_probesets)

oligo_probesets <- list(c("211836_s_at","214650_x_at"),"216617_s_at",
"207659_s_at",c("207323_s_at","209072_at"))
oligo_reference <- marker(expression, oligo_probesets)

micro_probesets <- list("204192_at", "203416_at")
micro_reference <- marker(expression, micro_probesets)

## Stepwise model selection for 2 transcripts (202429_s_at and
## 200850_s_at) and focusing on control samples (i.e. groups == 0)
lmlist <- swlm(expression[c("202429_s_at", "200850_s_at"),],
subset = which(groups == 0),
upper = formula(~neuron_reference + astro_reference +
oligo_reference + micro_reference))

pvalmat(lmlist, c("(Intercept)", "neuron_reference", "astro_reference",
"oligo_reference", "micro_reference"))
```

---

slt   *Extracts from a list of objects*

Description

Simple wrapper to extract a particular slot for each object in a list.

Usage

```
slt(ls,slot,index=NULL)
```

Arguments

ls	list of objects.
slot	character. Name of the slot to extract.
index	integer vector. Optional index vector to extract only a subset of the elements contained in the slot.

Details

The slot to be extracted is defined by matching "slot" against slot names of the first object in the list.

Slot extraction is done with sapply and the format of the output will depend on the content of the slots (see the value returned by sapply for details).

Value

sc	vector or matrix.

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

Examples

```
## Two arbitrary regression summaries in a list
x<-1:10
y1 <- x + rnorm(10,sd=1)
y2 <- x + rnorm(10,sd=5)

slm<-list()
slm[[1]] <- summary(lm(y1 ~ x))
slm[[2]] <- summary(lm(y2 ~ x))

## Extract R-squared from the list of summary.lm objects
slt(slm, "adj.r.squared")
```

---

swlm   *Stepwise model selection*

Description

Simple wrapper around stepAIC() (package MASS) to repeatedly perform stepwise model selection by AIC on several dependent variables (or responses, taken as rows of a matrix).

Usage

```
swlm(y,subset=NULL,upper,lower=formula(~1),direction='both',trace=FALSE,keep=NULL,verbose=FALSE)
```

Arguments

y	Numeric matrix (with responses as rows and samples as columns) or ExpressionSet. Typically the expression data with transcripts (i.e. for a microarray, probes) as rows and samples as columns. If an ExpressionSet is provided the expression data is extracted with the function exprs.
subset	Integer vector. Represents a subset of samples (specified as column indices in y) to use for model fitting. By default all samples are used.
verbose	logical. If TRUE (default) the response number being fitted is printed.
upper	see ?stepAIC
lower	see ?stepAIC
direction	see ?stepAIC
trace	see ?stepAIC
keep	see ?stepAIC

Details

The initial model for the stepwise approach only contains an intercept term.

Value

swft	List of stepwise-selected models (see ?stepAIC)

Author(s)

Alexandre Kuhn <alexandre.m.kuhn@gmail.com>

References

Kuhn A, Kumar A, Beilina A, Dillman A, Cookson MR, Singleton AB. Cell population-specific expression analysis of human cerebellum. BMC Genomics 2012, 13:610.

See Also

marker, lmfitst.

Examples

```
## Load example expression data (variable "expression")
## and phenotype data (variable "groups")
data("example")

## Four cell population-specific reference signals
neuron_probesets <- list(c("221805_at", "221801_x_at", "221916_at"),
"201313_at", "210040_at", "205737_at", "210432_s_at")
neuron_reference <- marker(expression, neuron_probesets)

astro_probesets <- list("203540_at",c("210068_s_at","210906_x_at"),
"201667_at")
astro_reference <- marker(expression, astro_probesets)

oligo_probesets <- list(c("211836_s_at","214650_x_at"),"216617_s_at",
"207659_s_at",c("207323_s_at","209072_at"))
oligo_reference <- marker(expression, oligo_probesets)

micro_probesets <- list("204192_at", "203416_at")
micro_reference <- marker(expression, micro_probesets)

## Stepwise model selection for 2 transcripts (202429_s_at and 200850_s_at)
## and focusing on control samples (i.e. groups == 0)
swlm(expression[c("202429_s_at", "200850_s_at"),],
subset = which(groups == 0),
upper = formula(~neuron_reference + astro_reference +
oligo_reference + micro_reference))
```

Index