Package ‘plw’

April 26, 2017

Type Package

Title Probe level Locally moderated Weighted t-tests.

Version 1.36.0

Date 2009-07-22

Author Magnus Astrand

Maintainer Magnus Astrand <magnus.astrand@astrazeneca.com>

Description Probe level Locally moderated Weighted median-t (PLW) and Locally Moderated Weighted-t (LMW).

Depends R (>= 2.10), affy (>= 1.23.4)

Imports MASS, affy, graphics, splines, stats

Suggests limma

License GPL-2

biocViews Microarray, OneChannel, TwoChannel, DifferentialExpression

NeedsCompilation yes

R topics documented:

plw-package .................................................. 2
AffySpikeU95Subset .......................................... 2
estimateMVbeta ............................................. 3
estimateSigma ................................................ 4
estimateSigmaMV ............................................ 6
estimateSigmaMVbeta ...................................... 7
getKnots ...................................................... 9
HowToPLW .................................................. 10
lmw .......................................................... 10
logitT ......................................................... 13
plw ........................................................... 14
scaleParameterPlot ......................................... 17
statByIndex .................................................. 19
topRankSummary ........................................... 21
varHistPlot ................................................ 22

Index 23
Description

Locally moderated weighted analysis of microarray data, at the probe level (PLW) or at the level of expression indexes (LMW).

Author(s)

Magnus Åstrand

References


See Also

plw, lwm

AffySpikeU95Subset

Spike-in data

Description

This AffyBatch-class represents part of the Affymetrix spike-in data set.

Usage

data(AffySpikeU95Subset)

Format

An AffyBatch-class containing 6 arrays.

Source

Array 1, 2, 21, 22, 41, and 42 of the Affymetrix U95 Latin square data set, thus the 6 arrays of group A and B, and a random subset of 1000 probe-sets together with the 16 spike-in probe-sets.
**estimateMVbeta**

Zero mean multivariate t-dist. with covariate dependent scale.

**Description**

Estimate the parameters $m$ and $\nu$ of the multivariate t-distribution with zero expectation, where $\nu$ is modeled as smooth function of a covariate. The covariance matrix $\Sigma$ is assumed to be known.

**Usage**

```r
estimateMVbeta(y, x, Sigma, maxIter = 200, epsilon = 1e-06, 
                 verbose = FALSE, nknots = 10, nOut = 2000, nIn = 4000, 
                 iterInit = 3, br = NULL)
```

**Arguments**

- `y`: Data matrix
- `x`: Covariate vector
- `Sigma`: Covariance matrix
- `maxIter`: Maximum number of iterations
- `epsilon`: Convergence criterion
- `verbose`: Print computation info or not
- `nknots`: Number of knots of spline for $\nu$
- `nOut`: Parameter for calculating knots, see `getKnots`
- `nIn`: Parameter for calculating knots, see `getKnots`
- `iterInit`: Number of iteration in when initiating $\Sigma$
- `br`: Knots, overrides nknots, n.out and n.in

**Details**

The multivariate t-distribution is parametrized as:

\[
y | c \sim N(\mu, c\Sigma)
\]

\[
c \sim \text{InvGamma}(m/2, m\nu/2)
\]

where $\nu$ is function of the covariate $x$: $\nu(x)$ and $N$ denotes a multivariate normal distribution, $\Sigma$ is a covariance matrix and $\text{InvGamma}(\alpha, \beta)$ is the inverse-gamma distribution with density function

\[
f(x) = (\beta)^\alpha \exp\{-\beta/x\}x^{-\alpha-1}/\Gamma(\alpha)
\]

A cubic spline is used to parameterize the smooth function $\nu(x)$

\[
\nu(x) = \exp\{H(x)^T \beta\}
\]

where $H : R \rightarrow R^{2p-1}$ is a set B-spline basis functions for a given set of $p$ interior spline-knots, see chapter 5 of Hastie (2001). In this application $\mu$ equals zero, and $m$ is the degrees of freedom.
**estimateSigma**

**Value**

- `Sigma`  The input covariance matrix for `y`
- `m`  Estimated shape parameter for inverse-gamma prior for gene variances
- `v`  Estimated scale parameter curve for inverse-gamma prior for gene variances
- `converged`  TRUE if the EM algorithms converged
- `iter`  Number of iterations
- `modS2`  Moderated estimator of gene-specific variances
- `histLogS2`  Histogram of log(s2) where s2 is the ordinary variance estimator
- `fittedDensityLogS2`  The fitted density for log(s2)
- `logs2`  Variance estimators, logged with base 2.
- `beta`  Estimated parameter vector $\beta$ of spline for $\nu(x)$
- `knots`  The knots used in spline for $\nu(x)$
- `x`  The input vector covariate vector `x`

**Author(s)**

Magnus Åstrand

**References**


**See Also**

plw, lmw, estimateSigmaMVbeta

---

**Description**

Estimate the covariance matrix $\Sigma$ of the multivariate t-distribution with zero expectation assuming the degrees of freedom is known.

**Usage**

```
estimateSigma(y, m, v, maxIter = 100, epsilon = 1e-06, verbose = FALSE)
```
estimateSigma

Arguments

- **y**: data matrix
- **m**: degrees of freedom
- **v**: scale parameter
- **maxIter**: maximum number of iterations
- **epsilon**: convergence criteria
- **verbose**: print computation info or not

Details

The multivariate t-distribution is parametrized as:

\[ y | c \sim N(\mu, c \Sigma) \]
\[ c \sim \text{InvGamma}(m/2, m \nu/2) \]

Here \( N \) denotes a multivariate normal distribution, \( \Sigma \) is a covariance matrix and \( \text{InvGamma}(\alpha, \beta) \) is the inverse-gamma distribution with density function

\[ f(x) = (\beta)^\alpha \exp\{-\beta/x\}x^{-\alpha-1}/\Gamma(\alpha) \]

In this application \( \mu \) equals zero, and \( m \) is the degrees of freedom.

Value

- **Sigma**: Estimated covariance matrix for \( y \)
- **iter**: Number of iterations

Author(s)

Magnus Åstrand

References


See Also

- estimateSigmaMV
estimateSigmaMV

Fit zero mean multivariate t-distribution

Description

estimate the parameters $\Sigma$, m and $\nu$ of the multivariate t-distribution with zero expectation.

Usage

estimateSigmaMV(y,maxIter=100,epsilon=0.000001,verbose=FALSE)

Arguments

y data matrix
maxIter maximum number of iterations
epsilon convergence criteria
verbose print computation info or not

Details

The multivariate t-distribution is parametrized as:

$$y|c \sim N(\mu, c\Sigma)$$

$$c \sim \text{InvGamma}(m/2, m\nu/2)$$

Here $N$ denotes a multivariate normal distribution, $\Sigma$ is a covariance matrix and $\text{InvGamma}(\alpha, \beta)$ is the inverse-gamma distribution with density function

$$f(x) = (\beta)^\alpha \exp\{-\beta/x\}x^{-\alpha-1}/\Gamma(\alpha)$$

In this application $\mu$ equals zero, and m is the degrees of freedom.

Value

Sigma Estimated covariance matrix for y
m Estimated shape parameter for inverse-gamma prior for gene variances
v Estimated scale parameter for inverse-gamma prior for gene variances
converged T if the EM algorithms converged
iter Number of iterations
modS2 Moderated estimator of gene-specific variances
histLogS2 Histogram of log(s2) where s2 is the ordinary variance estimator
fittedDensityLogS2 The fitted density for log(s2)

Author(s)

Magnus Åstrand
estimateSigmaMVbeta

References


See Also

estimateSigma

---

estimateSigmaMVbeta  Zero mean multivariate t-dist. with covariate dependent scale.

Description

Estimate the parameters $\Sigma$, $m$ and $\nu$ of the multivariate t-distribution with zero expectation, where $\nu$ is modeled as smooth function of a covariate.

Usage

```
estimateSigmaMVbeta(y, x, maxIter = 200, epsilon = 1e-06, verbose = FALSE, nknots = 10, nOut = 2000, nIn = 4000, iterInit = 3, br = NULL)
```

Arguments

- `y`  Data matrix
- `x`  Covariate vector
- `maxIter`  Maximum number of iterations
- `epsilon`  Convergence criteria
- `verbose`  Print computation info or not
- `nknots`  Number of knots of spline for $\nu$
- `nOut`  Parameter for calculating knots, see `getKnots`
- `nIn`  Parameter for calculating knots, see `getKnots`
- `iterInit`  Number of iteration in when initiating $\Sigma$
- `br`  Knots, overrides nknots, n.out and n.in
The multivariate t-distribution is parametrized as:

\[ y | c \sim N(\mu, c\Sigma) \]
\[ c \sim \text{InvGamma}(m/2, m\nu/2) \]

where \( \nu \) is function of the covariate \( x: \nu(x) \) and \( N \) denotes a multivariate normal distribution, \( \Sigma \) is a covariance matrix and \( \text{InvGamma}(a, b) \) is the inverse-gamma distribution with density function

\[ f(x) = (b)^a \exp\left\{-b/x\right\} x^{-a-1}/\Gamma(a) \]

A cubic spline is used to parameterize the smooth function \( \nu(x) \)

\[ \nu(x) = \exp\{H(x)^T \beta\} \]

where \( H : R \to R^{2p-1} \) is a set B-spline basis functions for a given set of \( p \) interior spline-knots, see chapter 5 of Hastie et al. (2001). In this application \( \mu \) equals zero, and \( m \) is the degrees of freedom. For details about the model see Kristiansson et al. (2005), Åstrand et al. (2007a, 2007b).

**Value**

- **Sigma**: Estimated covariance matrix for \( y \)
- **m**: Estimated shape parameter for inverse-gamma prior for gene variances
- **v**: Estimated scale parameter curve for inverse-gamma prior for gene variances
- **converged**: \( T \) if the EM algorithms converged
- **iter**: Number of iterations
- **modS2**: Moderated estimator of gene-specific variances
- **histLogS2**: Histogram of log(s2) where s2 is the ordinary variance estimator
- **fittedDensityLogS2**: The fitted density for log(s2)
- **logs2**: Variance estimators, logged with base 2.
- **beta**: Estimated parameter vector \( \beta \) of spline for \( \nu(x) \)
- **knots**: The knots used in spline for \( \nu(x) \)
- **x**: The input vector covariate vector \( x \)

**Author(s)**

Magnus Åstrand

**References**


**See Also**

plw, lmw
getKnots

Spline-knots for plw and lmw

Description

Computes a set of nKnots interior knots (if possible) plus 2 boundary knots so that:

1) the nOut smallest and highest data points (in x) lies below and above the lower and upper boundary knots respectively.
2) there is at least nIn data points between all knots.

Usage

getKnots(x, nKnots=10, nOut=2000, nIn=4000)

Arguments

x             Data vector
nKnots        Number of interior knots
nOut          Number of data points below and above the lower and upper boundary knots respectively.
nIn           Number of data points between knots.

Details

See the definition (R-code) for details.

Value

A vector of knots.

Author(s)

Magnus Åstrand

See Also

plw, lmw, estimateSigmaMVbeta
Description
Finds the location of the vignette HowToPLW and optionally opens it.

Usage
HowToPLW(view=TRUE)

Arguments
view logical, should the document be opened using the default PDF document reader?

Details
If the operating system is other than Windows, then the PDF viewer used is that given by Sys.getenv("R_PDFVIEWER"). The PDF viewer can be changed using Sys.putenv(R_PDFVIEWER=).
This function is used by drop-down Vignettes menu when the Rgui interface for Windows is used.

Value
Character string giving the file location.

Author(s)
Magnus Astrand

See Also
vignette, openPDF, openVignette, Sys.getenv, Sys.putenv

Examples
HowToPLW(view=FALSE)

Description
Computes Locally Moderated Weighted t-test for microarray data.

Usage
lmw(x,design=rep(1,ncol(x)),contrast=matrix(1), meanX=NULL, maxIter = 200, epsilon = 1e-06, verbose = TRUE, nknots = 10, nOut = 2000, nIn = 4000, knots = NULL, checkRegulation = TRUE)
Arguments

- `x`: Data, log2 expression indexes.
- `design`: design matrix
- `contrast`: contrast matrix
- `meanX`: Covariate used to model scale parameter, default=NULL (see details)
- `maxIter`: maximum number of iterations
- `epsilon`: convergence criteria
- `verbose`: print computation info or not
- `nknots`: Number of knots of spline for ν
- `nOut`: Parameter for calculating knots, see getKnots
- `nIn`: Parameter for calculating knots, see getKnots
- `knots`: Knots, if not NULL it overrides nknots, nOut and nIn
- `checkRegulation`: If TRUE, data is checked for a correct specified contrast (see details)

Details

This function computes the Locally Moderated Weighted-t statistic (LMW) described in Åstrand (2007b), thus calculating locally moderated weighted t-statistic, p-value and log2(FC) for each row of the data matrix `x`.

Each gene `g` (row of `x`) is modeled as:

\[
y_g | c_g \sim N(\mu_g, c_g \Sigma)
\]

\[
c_g \sim \text{InvGamma}(m/2, m\nu/2)
\]

where \(\nu\) is function of the mean intensity: \(\nu(\bar{\mu}_g)\), \(N\) denotes a multivariate normal distribution, \(\Sigma\) is a covariance matrix and InvGamma\((a, b)\) is the inverse-gamma distribution with density function

\[
f(x) = (b)^a \exp\{-b/x\} x^{-a-1}/\Gamma(a)
\]

Given the design matrix \(D\), \(\mu_g\) equals \(D\gamma_g\), and given the contrast matrix \(C\) the hypothesis \(C\gamma_g = 0\) is tested. \(C\) should be a one row matrix of same length as the column vector \(\gamma_g\).

See examples on how to specify the design and contrast matrices.

A cubic spline is used to parameterize the smooth function \(\nu(x)\)

\[
\nu(x) = \exp\{H(x)^T \beta\}
\]

where \(H : R \rightarrow R^{2p-1}\) is a set B-spline basis functions for a given set of \(p\) interior spline-knots, see chapter 5 of Hastie et al. (2001).

For details about the model see Kristiansson et al. (2005), Åstrand et al. (2007a,2007b).

As specified above, \(\nu\) is modeled as a function of mean intensity: \(\nu(\bar{\mu}_g)\). If the parameter meanX is not NULL, meanX is used instead of the mean intensity when modeling \(\nu\). Thus, if meanX is not NULL, meanX must be a vector of length equal to the number of rows of the data matrix `x`.

The parameter estimation procedure is based on the assumption that the specified contrast is close to zero for most genes, or at least that the median contrast over all genes is close to zero. A check is run on data to validate this assumptions. If the checking fails, with the error message "warning: most genes appears to be regulated..." and if YOU ARE SURE that the design and contrast is correct, use checkRegulation=FALSE.
Value

Sigma  Estimated covariance matrix for \( y = P^T x \)
m  Estimated shape parameter for inverse-gamma prior for gene variances
v  Estimated scale parameter curve for inverse-gamma prior for gene variances
converged  T if the EM algorithms converged
iter  Number of iterations
modS2  Moderated estimator of gene-specific variances
histLogS2  Histogram of log(s2) where s2 is the ordinary variance estimator
fittedDensityLogS2
  The fitted density for log(s2)
logs2  Variance estimators, logged with base 2.
t  Moderated t-statistic
coefficients  Estimated contrast
p.value  P-value from the moderated t-statistic
dfT  Degrees of freedom of the moderated t-statistic
weights  Weights for estimating the contrast
P  Transformation matrix
beta  Estimated parameter vector \( \beta \) of spline for \( \nu(x) \)
knots  The knots used in spline for \( \nu(x) \)

Author(s)

Magnus Åstrand

References


See Also

estimateSigmaMVbeta, plw

Examples

# ------------------------------------------
# Example analyzing the 6 arrays in the
# AffySpikeU95Subset data set
# Loading the data
data(AffySpikeU95Subset)
# Defining design and contrast matrix
group<-factor(rep(1:2,each=3))
design<-model.matrix(~group-1)
contrast<-matrix(c(1,-1),1,2)

# Computing RMA expression index
data.rma<-exprs(rma(AffySpikeU95Subset))

# Analyzing
model1<-lmw(data.rma,design=design,contrast=contrast,epsilon=0.01)

## Look at fitted vs observed density for log(s2)
varHistPlot(model1)

## Look at fitted curve for scale parameter
scaleParameterPlot(model1)

---

**logitT**  
*logit-t and t-test by row*

### Description
Functions for the logit-t test (Lemon et al. 2003) and the ordinary t-test computed for each row of an matrix.

### Usage

```r
logitTTransform(pm)  
logitTStat(affy.batch, group)  
studentTTest(x, group)
```

### Arguments

- `pm` A matrix of Pm intensities
- `affy.batch` An AffyBatch object
- `group` A group indicator vector, should have values 1 and 2 only.
- `x` A matrix

### Details
See the definition (R-code) of each function for details.

### Value

- `logitTTransform` returns a matrix
- `logitTStat` returns a vector with the logit-t statistic for each probe set.
- `studentTTest` returns a vector with t-statistic for each row of `x`.  

Author(s)
Magnus Åstrand

References

Examples
# ------------------------------------------
# Example analyzing the 6 arrays in the 
# AffySpikeU95Subset data set
# Loading the data
data(AffySpikeU95Subset)
# Vector with groups assignment
group<-factor(rep(1:2,each=3))
# logit-T statistic
logitT<-logitTStat(AffySpikeU95Subset, as.numeric(group))
# Computing RMA expression index
data.rma<exprs(rma(AffySpikeU95Subset))

# Ordinary t-test by row/gene
studentT<-studenttTTest(data.rma, as.numeric(group))

# Comparing genes ranked top-20
logitTTop20 <- rank(-abs(logitT)) < 21
studentTTop20<- rank(-abs(studentT)) < 21
table(logitTTop20,studentTTop20)

---

plw

**Probe level Locally moderated Weighted median-t.**

Description
Computes locally moderated weighted median t-test for microarray data.

Usage
```r
plw(x,design=rep(1,ncol(x)),contrast=matrix(1),
    probenames = unlist(ifelse(class(x) == "AffyBatch",
                        list(p = probeNames(x)),
                        list(p = NULL))),
    maxIter = 200, epsilon = 1e-06, verbose = TRUE,
    nknots = 10, nOut = 2000, nIn = 4000, knots = NULL,
    checkRegulation = TRUE)
```
Arguments

- **x**: Data, log2(PM) intensities or an AffyBatch object, see details.
- **design**: design matrix
- **contrast**: contrast matrix
- **probenames**: If not null, it is used to group PM probes into probe sets, see details.
- **maxIter**: maximum number of iterations
- **epsilon**: convergence criteria
- **verbose**: print computation info or not
- **nknots**: Number of knots of spline for $\nu$
- **nOut**: Parameter for calculating knots, see getKnots
- **nIn**: Parameter for calculating knots, see getKnots
- **knots**: Knots, if not NULL it overrides nknots, nOut and nIn
- **checkRegulation**: If TRUE, data is checked for a correct specified contrast (see details)

Details

This function computes the Probe level Locally moderated Weighted median-t statistic (PLW) as described in Åstrand (2007b), specially design for Affymetrix type data, or other microarray data with multiple probes.

The data object x should be either a matrix of perfect match (PM) intensities, or an object of class AffyBatch. When x is a matrix of PM intensities, the intensities should be background corrected, normalized, and logged (with base 2). If x is an AffyBatch object, the default background correction and normalization of RMA is applied to x.

When probenames is not null, it should be a vector of length equal to the number rows in the matrix x, giving the probe-set identity for each PM probe. Use the function probeNames in the affy package to get probenames when x is a matrix of log2(PM) intensities.

Inference is done for each PM probe, thus moderated t-statistic, p-value and log2(FC) is calculated for each probe. The median t-statistics for each probe-set is also computed.

Each PM probe g (row of x) is modeled as:

$$ y_g | c_g \sim N(\mu_g, c_g \Sigma) $$

$$ c_g \sim \text{InvGamma}(m/2, m\nu/2) $$

where $\nu$ is function of the mean intensity: $\nu(\bar{\mu}_g)$, $N$ denotes a multivariate normal distribution, $\Sigma$ is a covariance matrix and InvGamma$(a,b)$ is the inverse-gamma distribution with density function

$$ f(x) = (b)^a \exp\left(-b/x\right)x^{-a-1}/\Gamma(a) $$

Given the design matrix $D$, $\mu_g$ equals $D\gamma_g$, and given the contrast matrix $C$ the hypothesis $C\gamma_g = 0$ is tested. $C$ should be a one row matrix of same length as the column vector $\gamma_g$.

See examples on how to specify the design and contrast matrices.

A cubic spline is used to parameterize the smooth function $\nu(x)$

$$ \nu(x) = \exp\{H(x)^T \beta\} $$

where $H : R \rightarrow R^{2p-1}$ is a set B-spline basis functions for a given set of $p$ interior spline-knots, see chapter 5 of Hastie et al. (2001).
The parameter estimation procedure is based on the assumption that the specified contrast is close to zero for most genes, or at least that the median contrast over all genes is close to zero. A check is run on data to validate this assumption. If the checking fails, with the error message "warning: most genes appears to be regulated..." and if YOU ARE SURE that the design and contrast is correct, use checkRegulation=FALSE.

Value

Sigma Estimated covariance matrix for $y = P^T x$

m Estimated shape parameter for inverse-gamma prior for probe variances

v Estimated scale parameter curve for inverse-gamma prior for probe variances

converged T if the EM algorithms converged

iter Number of iterations

modS2 Moderated estimator of probe-specific variances

histLogS2 Histogram of log(s2) where s2 is the ordinary variance estimator

fittedDensityLogS2 The fitted density for log(s2)

logs2 Variance estimators, logged with base 2.

medianT Median moderated t-statistic for each probe-set

t Moderated t-statistic for each PM probe

coefficients Estimated contrast for each PM probe

p.value P-value from the moderated t-statistic for each PM probe

dfT Degrees of freedom of the moderated t-statistic

weights Weights for estimating the contrast

P Transformation matrix

beta Estimated parameter vector $\beta$ of spline for $\nu(x)$

knots The knots used in spline for $\nu(x)$

x The input vector covariate vector x

Author(s)

Magnus Åstrand

References


See Also

estimateSigmaMVbeta, lmw
Examples

# Example analyzing the 6 arrays in the
# AffySpikeU95Subset data set

# Loading the data
data(AffySpikeU95Subset)

# Defining design and contrast matrix
group<-factor(rep(1:2,each=3))
design<-model.matrix(~group-1)
contrast<-matrix(c(1,-1),1,2)

# Analyzing with an AffyBatch object as input
model1<-plw(AffySpikeU95Subset,design=design,contrast=contrast,
epsilon=0.01)

## Look at fitted vs observed density for log(s2)
varHistPlot(model1)

## Look at fitted curve for scale parameter
scaleParameterPlot(model1)

## Selecting top genes
topRankSummary(model1,nGenes=10)

## Plotting t-statistics and log2FC for top genes
par(mfrow=c(1,2))
plotSummaryT(model1,nGenes=20)
plotSummaryLog2FC(model1,nGenes=20)

### Analyzing with BG-adjusted and normalized PM data
pm1<-pm(bg.correct.rma(AffySpikeU95Subset, bgtype = 2))
pm2<-matrix(.C("qnorm_c", as.double(as.vector(pm1)),
as.integer(nrow(pm1)),
as.integer(ncol(pm1)))[[1]],
nrow(pm1),ncol(pm1))
data<-log2(pm2)
probeNames<-probeNames(AffySpikeU95Subset)
model2<-plw(data,design=design,contrast=contrast,
probeNames=probeNames,epsilon=0.01)

### Model1 and model2 should give identical result
# For example identical top ranking:
range(topRankSummary(model1)$t-
topRankSummary(model2)$t,na.rm=TRUE)

scaleParameterPlot Scale parameter plotted against mean intensity
scaleParameterPlot

Description

Will produce a scatter plot of variance estimators (logged) for each probe (probe set) against the corresponding mean intensity together with the fitted scale-parameter curve and points showing the knots of the used spline.

Usage

scaleParameterPlot(model,main="Scale parameter curve", col=1,pch=’.’,lty=1,curveCol=2,knotsPch=19,knotsCol=3)

Arguments

  model On object obtained from the function plw or lmw.
  main Main title of plot.
  col  Color for individual points (mean,logs2).
  pch  Plot symbol for individual points (mean,logs2).
  lty  Line type for fitted scale parameter curve.
  curveCol Line color for fitted scale parameter curve.
  knotsPch Plot symbol for spline knots.
  knotsCol Plot color for spline knots.

Author(s)

  Magnus Åstrand

See Also

  plw, lmw

Examples

# ------------------------------------------
# Example using the result of the analysis of # the 6 arrays in the AffySpikeU95Subset data set
# Loading the data
data(AffySpikeU95Subset)

# Defining design and contrast matrix
group<-factor(rep(1:2,each=3))
design<-model.matrix(~group-1)
contrast<-matrix(c(1,-1),1,2)

# Analyzing using plw
model1<-plw(AffySpikeU95Subset,design=design,contrast=contrast, epsilon=0.01)

## Look at fitted curve for scale parameter
scaleParameterPlot(model1)
DESCRIPTION

These functions give the same result as by(x, index, mad) by(x, index, mean) by(x, index, median) but are much faster. NOTE: The index vector is assumed to be SORTED and should contain INTEGER values only.

The function meanSdByRow computes mean and standard deviation for each row of the matrix mat. A list with mean and sd is returned and gives the same result as:

list(mean=apply(mat,1,mean),sd=apply(mat,1,sd))

USAGE

madByIndex(x, index)
meanByIndex(x, index)
medianByIndex(x, index)
orderStatByIndex(x, index, orderStat)
sdByIndex(x, index)
meanSdByRow(mat)

ARGUMENTS

x     Data vector
index Index vector
orderStat Which order statistic to compute
mat   Matrix

DETAILS

See the definition (R-code) of each function for details.

VALUE

All but the last function: A vector with the statistic for each level if index. meanSdByRow: A list with items mean and sd.

AUTHOR(S)

Magnus Åstrand

SEE ALSO

by, apply
## Example 1
### Computing, mad, mean and median by index.
### Compares with the result obtained using by(...)

```r
n<-10000
x<-rnorm(n)
index<-sort(round(runif(n,0.5,10.5)))
mad1<-madByIndex(x,index)
mad2<-by(x,index,mad)
mean1<-meanByIndex(x,index)
mean2<-by(x,index,mean)
median1<-medianByIndex(x,index)
median2<-by(x,index,median)
```

```r
par(mfrow=c(2,2),mar=c(4,4,1.5,.5),mgp=c(1.5,.25, 0))
plot(mad1,mad2,main="Comparing mad",pch=19)
abline(a=0,b=1,col=2)
plot(mean1,mean2,main="Comparing mean",pch=19)
abline(a=0,b=1,col=2)
plot(median1,median2,main="Comparing median",pch=19)
abline(a=0,b=1,col=2)
```

## Example 2
### Computing, median by index
### Compares with the running time when using by(...)

```r
n<-200000
x<-rnorm(n)
index<-sort(round(runif(n,0.5,10.5)))
system.time(median1<-medianByIndex(x,index))
system.time(median2<-by(x,index,median))
```

## Example 3
### Computing, mean and sd by row
### Compares with using apply

```r
nrow<-5000
ncol<-20
mat<-matrix(rnorm(ncol*nrow),nrow,ncol)
system.time(res1<-meanSdByRow(mat))
system.time(res2<-list(mean=apply(mat,1,mean),sd=apply(mat,1,sd)))
```

```r
par(mfrow=c(1,2),mar=c(4,4,1.5,.5),mgp=c(1.5,.25, 0))
plot(res1$mean,res2$mean,pch='.'
plot(res1$sd,res2$sd,pch='.'
```
Return or plots analysis result for top ranking or selected probe sets

Description

Returns (or plots) t-statistic and/or log2FC for each probe and median for each probe set. P

Usage

plotSummaryLog2FC(model, nGenes=50, genesOfRank=1:nGenes, genes=NULL)
plotSummaryT(model, nGenes=50, genesOfRank=1:nGenes, genes=NULL)
topRankSummary(model, nGenes=50, genesOfRank=1:nGenes, genes=NULL)

Arguments

model On object obtained from the function plw.
nGenes Gives summary for the nGenes top ranking genes
genesOfRank Gives summary for genes ranked genesOfRank
genes Gives summary for specific genes.

Author(s)

Magnus Åstrand

See Also

plw

Examples

# ------------------------------------------
# Example using the result of the analysis of
# the 6 arrays in the AffySpikeU95Subset data set

# Loading the data
data(AffySpikeU95Subset)

# Defining design and contrast matrix
group<-factor(rep(1:2,each=3))
design<-model.matrix(~group-1)
contrast<-matrix(c(1,-1),1,2)

# Analyzing using plw
model1<-plw(AffySpikeU95Subset,design=design,contrast=contrast, epsilon=0.01)

## Selecting top genes
topRankSummary(model1,nGenes=10)

## Plotting t-statistics and log2FC for top genes
par(mfrow=c(1,2))
plotSummaryT(model1,nGenes=20)
plotSummaryLog2FC(model1,nGenes=20)

varHistPlot

Variance histogram and density

Description
Will produce a histogram of observed variance estimators (logged) together with the fitted density.

Usage
varHistPlot(model,main="Histogram variance estimators",
           histCol=8,densityCol=2,drawLegend=TRUE)

Arguments
- **model**: On object obtained from the function plw or lmw.
- **main**: Main title of plot.
- **histCol**: Color for histogram bars.
- **densityCol**: Color for density function.
- **drawLegend**: To draw a legend or not.

Author(s)
Magnus Åstrand

See Also
plw, lmw

Examples

# Example using the result of the analysis of
# the 6 arrays in the AffySpikeU95Subset data set

data(AffySpikeU95Subset)

# Defining design and contrast matrix
group<-factor(rep(1:2,each=3))
design<-model.matrix(~group-1)
contrast<-matrix(c(1,-1),1,2)

# Analyzing using plw
model1<-plw(AffySpikeU95Subset,design=design,contrast=contrast, epsilon=0.01)

# Look at fitted vs observed density for log(s2)
varHistPlot(model1)
Index

*Topic **datasets**
  AffySpikeU95Subset, 2

*Topic **design**
  estimateSigma, 4
  estimateSigmaMV, 6
  lmw, 10
  plw, 14

*Topic **documentation**
  HowToPLW, 10

*Topic **htest**
  estimateMVbeta, 3
  estimateSigma, 4
  estimateSigmaMV, 6
  estimateSigmaMVbeta, 7
  lmw, 10
  plw, 14

*Topic **manip**
  getKnots, 9
  logitT, 13
  scaleParameterPlot, 17
  statByIndex, 19
  topRankSummary, 21
  varHistPlot, 22

*Topic **models**
  estimateMVbeta, 3
  estimateSigma, 4
  estimateSigmaMV, 6
  estimateSigmaMVbeta, 7
  lmw, 10
  plw, 14

*Topic **package**
  plw-package, 2

*Topic **univar**
  getKnots, 9
  logitT, 13
  scaleParameterPlot, 17
  statByIndex, 19
  topRankSummary, 21
  varHistPlot, 22

AffySpikeU95Subset, 2

estimateMVbeta, 3
estimateSigma, 4

estimateSigmaMV, 6
estimateSigmaMVbeta, 7

generateKnots, 9

HGU95subset1016 (AffySpikeU95Subset), 2
HowToPLW, 10

lmw, 10
logitT, 13
logitTStat (logitT), 13
logitTTransform (logitT), 13

madByIndex (statByIndex), 19
meanByIndex (statByIndex), 19
meanSdByRow (statByIndex), 19
medianByIndex (statByIndex), 19

openPDF, 10
openVignette, 10
orderStatByIndex (statByIndex), 19

plotSummaryLog2FC (topRankSummary), 21
plotSummaryT (topRankSummary), 21
plw, 14
plw-package, 2

scaleParameterPlot, 17
sdByIndex (statByIndex), 19
spikedProbesU95 (AffySpikeU95Subset), 2
statByIndex, 19
studenttTTest (logitT), 13
Sys.getenv, 10
Sys.putenv, 10

varHistPlot, 22
vignette, 10