LMGene

April 19, 2010

`genediff` *Raw p-value calculation function*

Description

Computes two vectors of p-values per gene or probe using gene-by-gene ANOVA with individual gene MSE using both the gene-specific MSE and the posterior mean MSE for each term in the ANOVA.
Assumes a fixed effects model and the correct denominator for all comparisons is the MSE.

Usage

```
genediff(eS, model=NULL)
```

Arguments

`eS`	Array data. must be an `ExpressionSet` object and the log-transformation and the normalization of `exprs(eS)` are recommended.
`model`	Model used for comparison; see details and `LMGene`.

Details

The argument `eS` must be an `ExpressionSet` object from the Biobase package. If you have a data in a `matrix` and information about the considered factors, then you can use `neweS` to convert the data into an `ExpressionSet` object. Please see `neweS` in more detail.

The `model` argument is an optional character string, constructed like the right-hand side of a formula for lm. It specifies which of the variables in the `ExpressionSet` will be used in the model and whether interaction terms will be included. If `model=NULL`, it uses all variables from the `ExpressionSet` without interactions. Be careful of using interaction terms with factors; this often leads to overfitting, which will yield an error.

Value

`pvlist`	a list containing two sets of p-values obtained by gene specific MSE and the posterior MSE methods.

Author(s)

David Rocke and Geun-Cheol Lee

References

David M. Rocke (2004), Design and analysis of experiments with high throughput biological assay data, Seminars in Cell & Developmental Biology, 15, 703-713.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

LMGene, rowaov

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)
LoggedSmpd0 <- neweS(lnorm(log(sample.mat)),vlist)

pvlist <- genediff(LoggedSmpd0)
pvlist$Posterior[1:5,]
```

`GetLMObj` *Function to get a simple lm object for a regression on the relevant model.*

Description

Internal to routines. Primarily used to get the X matrix corresponding to the model given (or the default model for the eS). Typically this is used to find residuals efficiently.

Usage

```
GetLMObj(eS, model=NULL)
```

Arguments

`eS`	An unprocessed `ExpressionSet` object.
`model`	Model used in the regression. Uses only variables from `pData(eS)`.

Value

Returns an lm object than corresponds to regressing one probe from the eS on the model specified (or the default model). See `lm`.

Author(s)

John Tillinghast

Examples

```
data(sample.eS)
lmod <- GetLMObj (sample.eS)
X <- lmod$x
```

`glog` *Generalized log transformation function*

Description

This function transforms the input values with the generalized log function.

Usage

```
glog(y, lambda)
```

Arguments

`y`	A matrix data
`lambda`	Parameter that should be determined

Details

Usually, matrix `y` is a modified matrix from an original matrix, after deducting parameter alpha. Lambda is one of the parameters that should be determined when using the glog function, and these parameters are decided by using the function `tranest`

Value

`yt`	A matrix containing a transformmed values by glog

Author(s)

David Rocke and Geun-Cheol Lee

References

Durbin, B.P., Hardin, J.S., Hawkins, D.M., and Rocke, D.M. (2002) A variance-stabilizing transformation for gene-expression microarray data, Bioinformatics, 18, S105–S110.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

`tranest`

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
sample.mat[1:5,1:4]

GloggedSmpd<-glog(sample.mat-50,500)
GloggedSmpd[1:5,1:4]
```

`jggrad2` *Generating Jacobian-corrected data*

Description

This function returns a Jacobian-corrected data with the given parameters lambda and alpha.

Usage

```
jggrad2(y, lambda, alpha)
```

Arguments

`y`	A matrix data containing array information
`lambda`	A parameter for glog transformation
`alpha`	A parameter for glog transformation

Details

The input arguments here would be rarely dealt by users directly.

Value

`data_matrix`	A matrix containing Jacobian-corrected data, gradient data by lambda and gradient data by alpha

Author(s)

David Rocke and Geun-Cheol Lee

References

B. Durbin and D.M. Rocke, (2003) Estimation of Transformation Parameters for Microarray Data, Bioinformatics, 19, 1360-1367.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

`msecalc`

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
dim(sample.mat)

JCSmpd<-jggrad2(sample.mat, 500, 50)
dim(JCSmpd)
```

`jglog` *Glog*

Description

Another Glog function

Usage

```
jglog(y, lambda)
```

Arguments

`y`	A matrix data
`lambda`	Parameter that should be determined

Details

Usually, matrix `y` is a modified matrix from an original matrix, after deducting parameter alpha. Lambda is one of the parameters that should be determined when using the glog function, and these parameters are decided by using the function `tranest`

Value

`yl`	A matrix containing a transformmed values by glog

Author(s)

David Rocke and Geun-Cheol Lee

References

Durbin, B.P., Hardin, J.S., Hawkins, D.M., and Rocke, D.M. (2002) A variance-stabilizing transformation for gene-expression microarray data, Bioinformatics, 18, S105–S110.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

`tranest`

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
sample.mat[1:5,1:4]

GloggedSmpd<-glog(sample.mat-50,500)
GloggedSmpd[1:5,1:4]
```

LMGene *LMGene main function*

Description

LMGene calls function genediff to calculate the raw p-values of all genes and then calls function pvadjust to calculate the adjusted p-values of all genes. Finally, calls function rowlist to list the names of genes that are selected as significant under the specified significance level.

Usage

```
LMGene(eS, model=NULL, level = 0.05)
```

Arguments

eS	Array data. must be an ExpressionSet object and the log-transformation and the normalization of exprs(eS) are recommended.
model	Specifies model to be used. Default is to use all variables from eS without interactions. See details.
level	Significance level

Details

The argument eS must be an ExpressionSet object from the Biobase package. If you have a data in a matrix and information about the considered factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS in more detail.

The level argument indicates False Discovery Rate, e.g. level=0.05 means 5

The model argument is an optional character string, constructed like the right-hand side of a formula for lm. It specifies which of the variables in the ExpressionSet will be used in the model and whether interaction terms will be included. If model=NULL, it uses all variables from the ExpressionSet without interactions. Be careful of using interaction terms with factors; this often leads to overfitting, which will yield an error.

Value

lmres	A list which contains significant gene names for each considered factor.

Author(s)

David Rocke and Geun-Cheol Lee

References

David M. Rocke (2004), Design and analysis of experiments with high throughput biological assay data, Seminars in Cell & Developmental Biology, 15, 703-713.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

genediff, pvadjust, rowlist

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)
LoggedSample <- neweS(lnorm(log(sample.mat)),vlist)

siggeneslist <- LMGene(LoggedSample, 'patient + dose', 0.01)
```

---

lnormeS *Function to apply lowessnorm to a transformed expression set. Returns the normalized expression set.*

Description

Basically the same as lnorm, but it applies to, and returns, expression sets instead of matrices.

Usage

```
lnormeS(eS, span=0.1)
```

Arguments

eS	A transformed expression set.
span	A parameter for lowess.

Value

Returns an expression set with the same vlist as eS, but the matrix has been normalized by lnorm.

Author(s)

John Tillinghast

References

http://www.idav.ucdavis.edu/~dmrocke/

See Also

lnorm, norm

Examples

```
data(sample.eS)
transeS (sample.eS, 667, 65) -> trsample.eS
lnormeS (trsample.eS) -> normtrsample.eS
```

---

lnorm *Lowess normalization function*

Description

Lowess normalization function

Usage

```
lnorm(mat1, span = 0.1)
```

Arguments

mat1	A matrix data to be normalized
span	A parameter for lowess

Details

mat1 must be a nbyp matrix, where n is the number of genes and p is the number of expression levels for each gene.

Value

matnorm1	Normalized matrix

Author(s)

David Rocke and Geun-Cheol Lee

References

http://www.idav.ucdavis.edu/~dmrocke/

See Also

norm

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
LoggedSmpd<-lnorm(log(sample.mat))
```

`mlm2lm` *Linear Model converting function*

Description

This function rule out the specified 'lm' class data out of the given 'c("mlm", "lm")' class data.

Usage

```
mlm2lm(lmobj, i)
```

Arguments

`lmobj`	An object of class 'c("mlm", "lm")'.
`i`	A specific number that indicates a 'lm' in `lmobj`.

Details

In case of multiple response from 'lm' function, this function can used.

Value

`lmobj2`	Selected 'lm' class data.

Author(s)

David Rocke and Geun-Cheol Lee

References

http://www.idav.ucdavis.edu/~dmrocke/

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.eS)
Smpd0 <- sample.eS
# model information
for(i in 1:length(varLabels(Smpd0))) {
```

```
  assign(paste('x', i, sep=''), as.factor(pData(Smpd0)[,i]))
}

fchar <- ''
for(i in 1:length(varLabels(Smpd0))) {
  fchar <- paste(fchar, paste('x', i, sep=''), ifelse(i<length(varLabels(Smpd0)), '+', ''
}
fchar2 <- paste("y ~",fchar)

# run regression and ANOVAs
y <- t(as.matrix(exprs(Smpd0)))
formobj <- as.formula(fchar2)
tmp <- lm(formobj)
class(tmp)

tmp2 <- mlm2lm(tmp, i)
class(tmp2)
```

`msa` *Relative mean square calculation function*

Description

Calculate the relative mean square values.

Usage

```
msa(v)
```

Arguments

`v`	A vector containing mean square values of all the factors.

Value

`rv`	relative mean square values for all factors.

Author(s)

David Rocke and Geun-Cheol Lee

References

http://www.idav.ucdavis.edu/~dmrocke/

Examples

```
#library
library(Biobase)
library(LMGene)

#data
#data
data(sample.eS)
```

```
Smpd0 <- sample.eS
# model information
for(i in 1:length(varLabels(Smpd0))){
  assign(paste('x', i, sep=''), as.factor(pData(Smpd0)[,i]))
}

fchar <- ''
for(i in 1:length(varLabels(Smpd0))){
  fchar <- paste(fchar, paste('x', i, sep=''), ifelse(i<length(varLabels(Smpd0)), '+', ''
}
fchar2 <- paste("y ~", fchar)

# run regression and ANOVAs
y <- t(as.matrix(exprs(Smpd0)))
formobj <- as.formula(fchar2)
tmp <- lm(formobj)
tmp2 <- mlm2lm(tmp, i)
tmp3 <- anova(tmp2)$Mean
tmp4 <- msa(tmp3)
rbind(tmp3, tmp4)
```

`msecalcmult` *MSE calculation function*

Description

Computes the mean square error and gradient for the global ANOVA.

Usage

```
msecalcmult(eS, lam, alpha, lowessnorm=FALSE, R, grads=TRUE)
```

Arguments

`eS`	Array data. must be an `ExpressionSet` object.
`lam`	A parameter for glog transformation.
`alpha`	A parameter for glog transformation.
`lowessnorm`	TRUE, if lowess method is going to be used.
`R`	The residual matrix, i.e., identity minus the hat matrix.
`grads`	If TRUE, return gradient as well as error. Not used with some kinds of optimization.

Details

The argument `eS` must be an `ExpressionSet` object from the Biobase package. If you have a data in a `matrix` and information about the considered factors, then you can use `neweS` to convert the data into an `ExpressionSet` object. Please see `neweS` in more detail.

Value

`msev`	A vector which contains MSE and gradient of two parameters.

Author(s)

David Rocke and Geun-Cheol Lee

References

B. Durbin and D.M. Rocke, (2003) Estimation of Transformation Parameters for Microarray Data, Bioinformatics, 19, 1360-1367.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

jggrad2, tranest2

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.eS)

lmod <- GetLMObj(sample.eS)
X <- lmod$x
U <- svd(X)$u
H <- crossprod(t(U), t(U))
n <- dim(H)[1]
R <- diag(rep(1,n)) - H

msecalc(sample.eS,500,50, FALSE, R)
```

---

msecalc *MSE calculation function*

Description

Computes the mean square error and gradient for the global ANOVA.

Usage

```
msecalc(eS, lam, alpha, lowessnorm, R)
```

Arguments

eS	Array data. must be an ExpressionSet object.
lam	A parameter for glog transformation.
alpha	A parameter for glog transformation.
lowessnorm	TRUE, if lowess method is going to be used.
R	The residual matrix, i.e., identity minus the hat matrix.

Details

The argument eS must be an ExpressionSet object from the Biobase package. If you have a data in a matrix and information about the considered factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS in more detail.

Value

msev A vector which contains MSE and gradient of two parameters.

Author(s)

David Rocke and Geun-Cheol Lee

References

B. Durbin and D.M. Rocke, (2003) Estimation of Transformation Parameters for Microarray Data, Bioinformatics, 19, 1360-1367.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

jggrad2, tranest2

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.eS)

lmod <- GetLMObj(sample.eS)
X <- lmod$x
U <- svd(X)$u
H <- crossprod(t(U), t(U))
n <- dim(H)[1]
R <- diag(rep(1,n)) - H

msecalc(sample.eS,500,50, FALSE, R)
```

---

neweS *Coercing to an ExpressionSet code*

Description

This function converts a matrix data and its experimental data into an object of 'ExpressionSet' class.

Usage

```
neweS(mat, vlist, vlabel = as.list(names(vlist)))
```

Arguments

mat	A matrix data to be converted.
vlist	A list which contains several factors representing the experiment description.
vlabel	A list of labels for the variables represented by the columns of pData of the 'ExpressionSet' object to be made.

Details

Must load Biobase package first for converting.
vlist contains all the considered factors in which level values of each element represent the corresponding column of mat.

Value

eset	The converted object of 'ExpressionSet' class.

Author(s)

David Rocke and Geun-Cheol Lee

References

http://www.idav.ucdavis.edu/~dmrocke/

See Also

as

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)

Smpdt <- neweS(sample.mat,vlist)

data(sample.eS)
identical(sample.eS, Smpdt)
```

---

norm *Additive normalization function*

Description

This function normalizes the matrix in additive way.

Usage

```
norm(mat1)
```

Arguments

mat1	A matrix data to be normalized

Value

matnorm	Normalized matrix

Author(s)

David Rocke and Geun-Cheol Lee

References

http://www.idav.ucdavis.edu/~dmrocke/

See Also

lnorm

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
LoggedSmpd<-norm(log(sample.mat))
```

---

psmeans *Function to take means of probesets.*

Description

This is used to estimate expression levels of genes based on the measurements for the relevant probes.

Usage

```
psmeans(eS, ind)
```

Arguments

eS	A transformed, normalized expression set.
ind	A vector used to indicate which probes go into which probesets.

Details

The vector ind has form like c(1,1,1,2,2,2,2,3,3,4,4,4,...) Each entry corresponds to one probe and tells the number of the probeset it belongs to.

Value

Returns an expression set with the same vlist as eS, but the matrix rows now correspond to probesets instead of individual probes.

Author(s)

John Tillinghast

Examples

```
data(sample.eS)
data(sample.ind)
transeS (sample.eS, 667, 65) -> trs.eS
lnormeS(trs.eS) -> ntrs.eS
psmeans (ntrs.eS, sample.ind) -> genesample.eS
```

---

pvadjust *P-value adjusting function*

Description

This function converts the given raw p-values into the FDR adjusted p-values using R package 'multtest'.

Usage

```
pvadjust(pvlist)
```

Arguments

pvlist	A list containing raw p-values

Details

pvlist is the output from genediff containing p-values from gene-specific MSE's and posterior MSE's.

Value

pvlist2	A list with the raw p-values and the newly computed FDR adjusted p-values

Author(s)

David Rocke and Geun-Cheol Lee

References

David M. Rocke (2004), Design and analysis of experiments with high throughput biological assay data, Seminars in Cell & Developmental Biology, 15, 703-713.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

genediff

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)
LoggedSmpd0<-neweS(lnorm(log(sample.mat)),vlist)

pvlist<-genediff(LoggedSmpd0)
pvlist$Posterior[1:5,]

apvlist<-pvadjust(pvlist)
names(apvlist)
apvlist$Posterior.FDR[1:5,]
```

---

rowaov *Gene by gene ANOVA function*

Description

Computes the mean squares and degrees of freedom for gene-by-gene ANOVAs.

Usage

```
rowaov(eS, model=NULL)
```

Arguments

eS	AArray data. must be an ExpressionSet object and the log-transformation and the normalization of exprs(eS) are recommended.
model	Model used for comparison. See details and LMGene.

Details

The argument eS must be an ExpressionSet object from the Biobase package. If you have a data in a matrix and information about the considered factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS in more detail.

The model argument is an optional character string, constructed like the right-hand side of a formula for lm. It specifies which of the variables in the ExpressionSet will be used in the model and whether interaction terms will be included. If model=NULL, it uses all variables from the ExpressionSet without interactions. Be careful of using interaction terms with factors; this often leads to overfitting, which will yield an error.

Value

resmat	A matrix of MSE and DF of all factors for all genes.

Author(s)

David Rocke and Geun-Cheol Lee

References

David M. Rocke (2004), Design and analysis of experiments with high throughput biological assay data, Seminars in Cell & Developmental Biology, 15, 703-713.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

genediff, mlm2lm

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)
LoggedSmpd0 <- neweS(lnorm(log(sample.mat)),vlist)

resmat <- rowaov(LoggedSmpd0)
resmat[,1:3]
```

---

rowlist *Gene name listing function*

Description

This function makes significant gene list for a specified factor, where genes are selected as significant by the given p-values and significance level.

Usage

```
rowlist(genemat, effnum, apvlist, level, posterior = TRUE)
```

Arguments

genemat	A matrix data of array.
effnum	Factor number.
apvlist	A vector with FDR adjusted p-value.
level	Significance level.
posterior	TRUE, if adjusted p-values are to be computed with Posterior method.

Details

genemat is an n-by-p matrix of expression values. effnum is the column number for the effect of interest. apvlist is a matrix of p-values from pvadjust or genediff the routine returns a list of genes whose FDR p-value is less than level using either individual gene or posterior MSE's. This function returns gene names if rownames(genemat) is not NULL, or gene numbers otherwise. level indicates False Discovery Rate. e.g.) level 0.05 means 5

Value

genelist — A vector containing gene names if rownames(genemat) is not NULL, or gene numbers otherwise.

Author(s)

David Rocke and Geun-Cheol Lee

References

David M. Rocke (2004), Design and analysis of experiments with high throughput biological assay data, Seminars in Cell & Developmental Biology, 15, 703-713.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

LMGene, rowaov

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)
LoggedSmpd0<-neweS(lnorm(log(sample.mat)), vlist)

pvlist <- genediff(LoggedSmpd0)
apvlist <- pvadjust(pvlist)

genelist <- rowlist(exprs(LoggedSmpd0), 2, apvlist, 0.01)
genelist
```

---

sample.eS — *Sample array data for LMGene*

Description

Sample 'ExpressionSet' class data.

Usage

```
data(sample.eS)
```

Format

Formal class 'ExpressionSet' [package "Biobase"].

Details

identical with 'neweS(sample.mat, vlist)'

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)

Smpdt <- neweS(sample.mat,vlist)

data(sample.eS)
identical(sample.eS, Smpdt)
```

---

sample.ind *Sample probeset index vector*

Description

Vector indicating which probeset each probe belongs to

Usage

```
data(sample.ind)
```

Format

A vector of integers, e.g., c(1,1,1,2,2,3,3,3,4,4,...). Length is of course equal to the number of probes (rows) in sample.mat.

Examples

```
data(sample.eS)
data(sample.ind)
transeS (sample.eS, 667, 65) -> trs.eS
lnormeS(trs.eS) -> ntrs.eS
psmeans (ntrs.eS, sample.ind) -> genesample.eS
```

sample.mat *Sample array data for LMGene package*

Description

A matrix of array data

Usage

```
data(sample.mat)
```

Format

A data frame measuring 613 probes on the 32 samples.

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.mat)
data(vlist)

Smpdt<-neweS(sample.mat,vlist)

data(sample.eS)
identical(sample.eS, Smpdt)
```

tranest2 *Glog transformation parameter estimation function 2*

Description

A sub-function of `tranest` which search the best parameters for glog transformation.

Usage

```
tranest2(eS, starting = FALSE, lambda = 1000, alpha = 0, gradtol = 0.001, lowess
```

Arguments

`eS`	Array data. must be an `ExpressionSet` object.
`starting`	TRUE, if the given initial parameter values are used.
`lambda`	Initial parameter value for lambda.
`alpha`	Initial parameter value for alpha.
`gradtol`	a positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm.

lowessnorm	TRUE, if lowess method is going to be used.
method	Set optimization method; default is modified Gauss-Newton (nlm). See tranest.
model	Model in terms of vlist which is compared to transformed expression data. See tranest.

Details

The argument eS must be an ExpressionSet object from the Biobase package. If you have a data in a matrix and information about the considered factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS in more detail.

The model argument is an optional character string, constructed like the right-hand side of a formula for lm. It specifies which of the variables in the ExpressionSet will be used in the model and whether interaction terms will be included. If model=NULL, it uses all variables from the ExpressionSet without interactions. Be careful of using interaction terms with factors; this often leads to overfitting, which will yield an error.

Value

tranpar	A numeric vector containing the best parameter for 'lambda' and 'alpha'.

Author(s)

David Rocke and Geun-Cheol Lee

References

B. Durbin and D.M. Rocke, (2003) Estimation of Transformation Parameters for Microarray Data, Bioinformatics, 19, 1360-1367.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

jggrad2, tranest2

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.eS)

tranpar <- tranest2(sample.eS, lambda= 500, alpha=50)
tranpar
```

`tranestmult` *Glog transformation parameter estimation function for multiple parameters*

Description

A sub-function of `tranest` which searches the best parameters for glog transformation.

Usage

```
tranestmult (eS, starting = FALSE, lambda = 1000, alpha = 0, gradtol = 0.001, lo
```

Arguments

`eS`	Array data. must be an `ExpressionSet` object.
`starting`	TRUE, if the given initial parameter values are used.
`lambda`	Initial parameter value for lambda.
`alpha`	Initial parameter value for alpha.
`gradtol`	a positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm.
`lowessnorm`	TRUE, if lowess method is going to be used.
`method`	Set optimization method; default is modified Gauss-Newton (nlm). See `tranest`.
`max_iter`	Max. number of iterations of `nlm` to use in optimization.
`model`	Model in terms of vlist which is compared to transformed expression data. See `tranest`.

Details

This is primarily an internal function. The normal way of calling it would be to call `tranest` with the option mult=TRUE.

The argument `eS` must be an `ExpressionSet` object from the Biobase package. If you have a data in a `matrix` and information about the considered factors, then you can use `neweS` to convert the data into an `ExpressionSet` object. Please see `neweS` in more detail.

The `model` argument is an optional character string, constructed like the right-hand side of a formula for lm. It specifies which of the variables in the `ExpressionSet` will be used in the model and whether interaction terms will be included. If `model=NULL`, it uses all variables from the `ExpressionSet` without interactions. Be careful of using interaction terms with factors; this often leads to overfitting, which will yield an error.

Value

`tranpar`	A list (not a vector) containing the best parameter for 'lambda' and the best vector for 'alpha'.

Author(s)

David Rocke and Geun-Cheol Lee

References

B. Durbin and D.M. Rocke, (2003) Estimation of Transformation Parameters for Microarray Data, Bioinformatics, 19, 1360-1367.

http://www.idav.ucdavis.edu/~dmrocke/

See Also

tranest, tranest2

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.eS)

tranpar <- tranestmult(sample.eS, lambda= 500, alpha=50)
tranpar
```

---

tranest *Glog transformation parameter estimation function*

Description

Finds the best parameters for glog transformation.

Usage

```
tranest(eS, ngenes = -1, starting = FALSE, lambda = 1000, alpha = 0, gradtol = 0
```

Arguments

eS	Array data. must be an ExpressionSet object.
ngenes	Number of genes that is going to be used for the parameter estimation.
starting	TRUE, if the given initial parameter values are used.
lambda	Initial parameter value for lambda.
alpha	Initial parameter value for alpha.
gradtol	a positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm.
lowessnorm	TRUE, if lowess method is going to be used.
method	Determines optimization method. Default is 1, which corresponds to a Newton-type method (see nlm). Method 2 is based on the Nelder-Mead method (see optim).
mult	If true, tranest will use a vector alpha with one entry per sample. Default is false (same alpha for every sample).
model	Specifies model to be used. Default is to use all variables from eS without interactions. See details.

Details

The argument eS must be an ExpressionSet object from the Biobase package. If you have a data in a matrix and information about the considered factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS in more detail.

The model argument is an optional character string, constructed like the right-hand side of a formula for lm. It specifies which of the variables in the ExpressionSet will be used in the model and whether interaction terms will be included. If model=NULL, it uses all variables from the ExpressionSet without interactions. Be careful of using interaction terms with factors; this often leads to overfitting, which will yield an error.

Value

tranpar A list containing the best parameter for 'lambda' and 'alpha'.

Author(s)

David Rocke, Geun-Cheol Lee and John Tillinghast

References

B. Durbin and D.M. Rocke, (2003) Estimation of Transformation Parameters for Microarray Data, Bioinformatics, 19, 1360-1367.

http://www.idav.ucdavis.edu/~dmrocke/

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(sample.eS)

tranpar <- tranest(sample.eS, 100)
tranpar
tranpar <- tranest(sample.eS, mult=TRUE)
tranpar
```

---

transeS *Function to apply the glog transform to an expression set. Returns the transformed expression set (not normalized).*

Description

For each element in the array of expression data, this applies the glog transform y -> glog (y-alpha, lambda). If alpha is a vector, it must have one entry per sample, and transeS will use the appropriate entry from the vector.

Usage

```
transeS(eS, lambda, alpha)
```

Arguments

eS	An unprocessed expression set.
lambda	The parameter lambda to be used in the glog transform (Durbin and Rocke 2003).
alpha	The alpha parameter(s) for the glog transform. May be a single number used for all samples, or a vector with one entry per sample.

Value

Returns an expression set with the same vlist as eS, but the matrix is now glog-transformed. That matrix can be normalized with norm or lnorm.

Author(s)

John Tillinghast

Examples

```
data(sample.eS)
transeS (sample.eS, 667, 65) -> trsample.eS
```

---

vlist *Sample experimental data for LMGene package*

Description

A list data representing experiment description information for the sample matrix array data, 'sample.mat'.

Usage

```
data(vlist)
```

Examples

```
#library
library(Biobase)
library(LMGene)

#data
data(vlist)

vlist
```

Index

∗Topic **datasets**
sample.eS, 19
sample.ind, 20
sample.mat, 21
vlist, 26
∗Topic **manip**
neweS, 13
rowlist, 18
∗Topic **math**
GetLMObj, 2
glog, 3
jggrad2, 4
jglog, 5
lnorm, 8
lnormeS, 7
msa, 10
msecalc, 12
msecalcmult, 11
norm, 14
psmeans, 15
tranest, 24
tranest2, 21
tranestmult, 23
transeS, 25
∗Topic **models**
genediff, 1
LMGene, 6
mlm2lm, 9
pvadjust, 16
rowaov, 17

as, *14*

genediff, 1, *6, 7, 17, 18*
GetLMObj, 2
glog, 3

jggrad2, 4, *12, 13*, 22
jglog, 5

lm, *2*
LMGene, *1, 2*, 6, *17*, *19*
lnorm, *7*, 8, *8*, *15*, *26*
lnormeS, 7

mlm2lm, 9, *18*
msa, 10
msecalc, *4*, 12
msecalcmult, 11

neweS, *1*, *6*, *11*, 13, *13*, *17*, *22, 23*, *25*
nlm, *23, 24*
norm, *8*, 14, *26*

optim, *24*

psmeans, 15
pvadjust, *6, 7*, 16

rowaov, *2*, 17, *19*
rowlist, *6, 7*, 18

sample.eS, 19
sample.ind, 20
sample.mat, 21

tranest, *3*, *5*, *22, 23*, 24, *24*
tranest2, *12, 13*, 21, *22*, *24*
tranestmult, 23
transeS, 25

vlist, 26