Package 'LMGene'

April 14, 2017

Version 2.30.0

Date 2013-07-24

Title LMGene Software for Data Transformation and Identification of Differentially Expressed Genes in Gene Expression Arrays

Author David Rocke, Geun Cheol Lee, John Tillinghast, Blythe Durbin-Johnson, and Shiquan Wu

Depends R (>= 2.10.0), Biobase (>= 2.5.5), multtest, survival, affy

Suggests affydata

Maintainer Blythe Durbin-Johnson <bpdurbin@ucdavis.edu>

Description LMGene package for analysis of microarray data using a linear model and glog data transformation

License LGPL

URL http://dmrocke.ucdavis.edu/software.html

biocViews Microarray, DifferentialExpression, Preprocessing

NeedsCompilation no

R topics documented:

---

genediff *Gene-by-gene and posterior p-value calculation function.*

Description

Computes two sets of p-values per gene or probe via gene-by-gene ANOVA, using both the gene-specific MSE and the posterior MSE for each term in the ANOVA. P-values are not adjusted for multiple testing.
Assumes a fixed effects model and that the correct denominator for all comparisons is the MSE.

Usage

```
genediff(eS, model = NULL, method = c("MLE", "MOM", "MOMlog"),
verbose = TRUE)
```

Arguments

eS	An ExpressionSet object. Any transformation and normalization of exprs(eS) should be conducted prior to use in genediff.
model	Model used for comparison; see details and LMGene.
method	Method by which posterior p-values are calculated. Default "MLE".
verbose	If TRUE, the prior degrees of freedom and mean reciprocal precision are printed. See details.

Details

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

The method argument specifies how the adjusted MSE and degrees of freedom should be calculated for use in computation of the posterior p-values:

"MLE" Default. Calculate adjusted MSE and degrees of freedom by maximum likelihood estimation, as described in Wright and Simon (2003).

"MOM" Calculate adjusted MSE and degrees of freedom by method of moments, as described in Rocke (2003).

"MOMlog" Calculate adjusted MSE and degrees of freedom by method of moments on log scale, as described in Smyth (2004). Uses functions fitFdist and trigammainverse from the package limma. Note that the method of Smyth (2004) is used here to calculate the posterior MSE, but not to directly calculate the posterior p-values.

All three methods assume that the gene-specific MSE's follow a gamma distribution with mean tau. (NB: Notation and parameterization vary somewhat between each of the source papers.) The mean of the gamma distribution, tau, is modeled with an inverse gamma prior with hyperparameters alpha and beta. Empirical Bayes methods are used to estimate the prior hyperparameters, either by maximum likelihood, method of moments, or method of moments on the log scale. The "posterior MSE" is the posterior mean of the variances given the observed gene-specific MSE's.

If `verbose = TRUE`, the function prints the estimated prior degrees of freedom, which equals twice the prior shape parameter alpha, and the estimated prior mean reciprocal precision, or 1/(alpha*beta).

All p-values are calculated from fixed-effects ANOVA F statistics, using either the gene-specific MSE or the posterior MSE as the denominator.

Value

A list with components:

`Gene.Specific`	A matrix of p-values calculated using the gene-specific MSE, with one row for each gene/probe and one column for each factor
`Posterior`	A matrix of p-values calculated using the posterior MSE, with one row for each gene/probe and one column for each factor

Author(s)

David Rocke, Geun-Cheol Lee, and Blythe Durbin-Johnson

References

Rocke, D.M. (2003) Heterogeneity of variance in gene expression microarray data. http://dmrocke.ucdavis.edu/papers/empbayes2.pdf

Smyth, G.K (2004) Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* **3**, Article 3. http://www.bepress.com/sagmb/vol3/iss1/art3/

Wright, G.W. and Simon, R.M. (2003) A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics* **19**, 2448–2455.

http://dmrocke.ucdavis.edu

See Also

LMGene, rowaov, neweS

Examples

```
library(Biobase)
library(LMGene)

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

raw.eS <- neweS(sample.mat, vlist)

# glog transform data
trans.eS <- transeS(raw.eS, lambda = 727, alpha = 56)
```

```
# calculate p-values
pvlist <- genediff(trans.eS)
pvlist$Posterior[1:5,]
```

`glog` *Generalized log transformation function*

Description

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

Usage

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

Arguments

`y`	A data matrix
`lambda`	Transformation parameter

Details

The glog transformation of a variable `y` is defined as `log(y + sqrt(y^2 + lambda))`. Using `lambda = 0` corresponds to the log transformation, up to a scale factor of 2. (Other, equivalent expressions exist for the glog transformation. See Durbin et al. (2002) and Huber et al. (2002) for futher details.)

The input matrix `y` may be modified prior to transformation by subtracting a constant or vector (`"alpha"`). The parameters `lambda` and `alpha` may be estimated from `tranest`.

Value

`yt`	A matrix of glog-transformed values

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.

Huber, W., Von Heydebreck, A., Sueltmann, H., Poustka, A., and Vingron, M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression, *Bioinformatics*, **18**, S96–S104.

http://dmrocke.ucdavis.edu

See Also

`tranest`, `transeS`

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 unadjusted gene-specific and posterior p-values of all genes and then calculates the FDR-adjusted p-values of all genes. Significant genes for each factor in model (based on either the gene-specific or posterior FDR-adjusted p-values) are output.

Usage

```
LMGene(eS, model = NULL, level = 0.05, posterior = FALSE,
method = c("MLE", "MOM", "MOMlog"))
```

Arguments

eS	An ExpressionSet object. Any transformation and normalization of exprs(eS) should be conducted prior to use in LMGene.
model	Specifies model to be used. Default is to use all variables from eS without interactions. See details.
level	Significance level
posterior	If TRUE, the posterior FDR-adjusted p-values are used in listing significant genes for each factor. Default is to use gene-specific FDR-adjusted p-values.
method	Method by which the posterior p-values are calculated. Default is "MLE".

Details

If you have data in a matrix and information about experimental design factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS for more detail.

The level argument indicates the False Discovery Rate, e.g. level=0.05 means a 5 percent FDR.

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.

See genediff for details of method.

Value

`lmres`	A list with one component for each factor in `model`. Each component consists of a character vector with one element per significant gene. If no genes are significant for a given factor, the component for that factor is set to `"No significant genes"`.

Author(s)

David Rocke and Geun-Cheol Lee

References

Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society, Series B*, **57**, 289–300.

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

http://dmrocke.ucdavis.edu

See Also

genediff, neweS

Examples

```
library(Biobase)
library(LMGene)

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

raw.eS <- neweS(sample.mat, vlist)

# glog transform data
trans.eS <- transeS(raw.eS, lambda = 727, alpha = 56)

# Identify significant genes, using an FDR of 1 percent
LMGene(trans.eS, level = 0.01)
```

---

lnorm *Lowess normalization function*

Description

Lowess normalization function

Usage

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

Arguments

`mat1`	A data matrix to be normalized
`span`	Lowess smoother span. Larger values give more smoothness.

Details

mat1 must be a p by n matrix, where p is the number of genes and n is the number of arrays or samples

Value

matnorm1	Normalized matrix

Author(s)

David Rocke and Geun-Cheol Lee

References

http://dmrocke.ucdavis.edu

See Also

lnormeS, norm

Examples

```
library(Biobase)
library(LMGene)

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

raw.eS <- neweS(sample.mat, vlist)

# glog transform data
trans.eS <- transeS(raw.eS, lambda = 727, alpha = 56)

# normalize
normed.exprs <- lnorm(exprs(trans.eS))
```

---

lnormeS *Function to apply lowess normalization to an expression set.*

Description

Like lnorm, but applies to and returns an ExpressionSet or AffyBatch object instead of a matrix.

Usage

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

Arguments

eS	An ExpressionSet or AffyBatch object
span	Smoothing parameter for lowess. Larger values correspond to more smoothness.

Value

Returns an ExpressionSet with exprs(eS) normalized by lnorm.

Author(s)

John Tillinghast, Blythe Durbin-Johnson

References

http://dmrocke.ucdavis.edu

See Also

lnorm, norm

Examples

```
library(LMGene)
library(Biobase)

data(sample.eS)

# glog transform expression set
trsample.eS <- transeS (sample.eS, 667, 65)

# normalize expression set
normtrsample.eS <- lnormeS (trsample.eS)
```

---

neweS *Coerce a matrix to class ExpressionSet*

Description

This function converts a data matrix into an ExpressionSet object.

Usage

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

Arguments

mat	A data matrix to be converted.
vlist	A list, each component of which describes a factor in the experimental design.
vlabel	A list of labels for each component of vlist.

Details

Each element of a component of vlist corresponds to a column of mat. See vlist for an example.

Value

eset	An ExpressionSet object.

Author(s)

David Rocke and Geun-Cheol Lee

References

http://dmrocke.ucdavis.edu

See Also

vlist

Examples

```
library(Biobase)
library(LMGene)

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

Smpdt <- neweS(sample.mat,vlist)

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

---

norm *Normalization function*

Description

This function normalizes a matrix by subtracting the column (sample) mean from each element and adding the grand mean.

Usage

```
norm(mat1)
```

Arguments

mat1	A matrix to be normalized

Value

matnorm	Normalized matrix

Author(s)

David Rocke and Geun-Cheol Lee

References

http://dmrocke.ucdavis.edu

See Also

lnorm

Examples

```
library(Biobase)
library(LMGene)

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

raw.eS <- neweS(sample.mat, vlist)

# glog transform data
trans.eS <- transeS(raw.eS, lambda = 727, alpha = 56)

# normalize
normed.exprs <- norm(exprs(trans.eS))
```

---

plotMeanSD *Plotting function for gene means and standard deviations*

Description

Plots the row standard deviation of a matrix of expression data against the row mean, or the rank of the row mean.

Usage

```
plotMeanSD(indata, by.rank = TRUE, line = FALSE, ymax = NULL)
```

Arguments

indata	An object of class matrix, data.frame, ExpressionSet, or AffyBatch
by.rank	If TRUE, the row standard deviations are plotted against the ranks of the row means. Otherwise, the row standard deviations are plotted against the row means themselves.
line	If TRUE, a lowess smoother line is drawn on the plot.
ymax	The upper limit for the plot y-axis. If missing, axis limits are generated automatically by plot.

Details

Generates a scatter plot of the row standard deviations of a matrix of expression data against the row means or ranks of the row means.

Value

NULL

Author(s)

Rachel Chen and Blythe Durbin-Johnson

Examples

```
library(LMGene)
library(Biobase)

data(sample.eS)
# transform data
trans.eS <- transeS(sample.eS, lambda = 727, alpha = 56)

# plot SD against rank of mean
plotMeanSD(trans.eS, line = TRUE)
plotMeanSD(sample.eS, line = TRUE, ymax = 1000)
```

---

psmeans *Function to take means of probesets.*

Description

Converts an ExpressionSet or AffyBatch object with one row of expression data per probeset into an ExpressionSet or AffyBatch object with one row per probe.

Usage

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

Arguments

eS	An ExpressionSet or AffyBatch object
ind	A vector used to indicate which probes go into which probesets.

Details

Each entry of ind corresponds to one probe and tells the number of the probeset it belongs to. See tranestAffyProbeLevel and sample.ind for examples.

Value

Returns an ExpressionSet or AffyBatch object with the expression matrix rows corresponding to probesets instead of individual probes. Elements of the returned ExpressionSet or AffyBatch object are means over each probeset.

Author(s)

John Tillinghast

See Also

tranestAffyProbeLevel, sample.ind

Examples

```
library(LMGene)
library(Biobase)

data(sample.eS)
data(sample.ind)

# glog transform data
trs.eS <- transeS (sample.eS, 667, 65)

# lowess normalize
ntrs.eS <- lnormeS(trs.eS)

# take means over probesets
genesample.eS<- psmeans (ntrs.eS, sample.ind)
```

`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	An ExpressionSet object. Any transformation and normalization of eS should be done prior to use in rowaov.
model	Model used for comparison. See details and LMGene.

Details

If you have data in a matrix and information about experimental design factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS for 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 MSEs and degrees of freedom for all model factors and all genes. The first rows of resmat contain MSE's for each effect in model, ending with the residual MSE. The remaining rows contain degrees of freedom for each effect in the model, ending with the residual d.f. Each column corresponds to a gene.

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://dmrocke.ucdavis.edu

See Also

genediff, LMGene

Examples

```
library(Biobase)
library(LMGene)

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

raw.eS <- neweS(sample.mat, vlist)

# glog transform data
trans.eS <- transeS(raw.eS, lambda = 727, alpha = 56)

# Perform gene-by-gene anova
resmat <- rowaov(trans.eS)
resmat[,1:3]
```

---

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)`, up to metadata

Examples

```
library(Biobase)
library(LMGene)

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

Smpdt <- neweS(sample.mat,vlist)

data(sample.eS)
identical(exprs(sample.eS), exprs(Smpdt))
identical(pData(sample.eS), pData(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 equal to the number of probes (rows) in sample.mat.

Examples

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

sample.mat *Sample array data for LMGene package*

Description

A matrix of array data

Usage

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

Format

A matrix measuring 613 rows (probes) by 32 columns (samples).

Examples

```
library(Biobase)
library(LMGene)

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

Smpdt<-neweS(sample.mat,vlist)

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

tranest *Glog transformation parameter estimation function*

Description

Estimates parameters for the glog transformation, by maximum likelihood or by minimizing the stability score.

Usage

```
tranest(eS, ngenes = -1, starting = FALSE, lambda = 1000, alpha = 0,
    gradtol = 1e-3, lowessnorm = FALSE, method=1, mult=FALSE, model=NULL,
SD = FALSE, rank = TRUE, model.based = TRUE, rep.arrays = NULL)
```

Arguments

eS	An ExpressionSet object
ngenes	Number of genes to be used in parameter estimation. Default is to use all genes unless there are more than 100,000, in which case a subset of 50,000 genes is selected at random.
starting	If TRUE, user-specified starting values for lambda and alpha are input to the optimization routine
lambda	Starting value for parameter lambda. Ignored unless starting = TRUE
alpha	Starting value for parameter alpha. Ignored unless starting = TRUE
gradtol	A positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm
lowessnorm	If TRUE, lowess normalization (using lnorm) is used in calculating the likelihood.
method	Determines optimization method. Default is 1, which corresponds to a Newton-type method (see nlm and details.)
mult	If TRUE, tranest will use a vector alpha with one (possibly different) entry per sample. Default is to use same alpha for every sample. SD and mult may not both be TRUE.
model	Specifies model to be used. Default is to use all variables from eS without interactions. See details.
SD	If TRUE, transformation parameters are estimated by minimizing the stability score rather than by maximum likelihood. See details.
rank	If TRUE, the stability score is calculated by regressing the replicate standard deviations on the ranks of the gene/row means (rather than on the means themselves). Ignored unless SD = TRUE
model.based	If TRUE, the stability score is calculated using the standard deviations of residuals from the linear model in model. Ignored unless SD = TRUE
rep.arrays	List of sets of replicate arrays. Each element of rep.arrays should be a vector with entries corresponding to arrays (columns) in exprs(eS) conducted under the same experimental conditions, i.e., with identical rows in pData(eS). Ignored unless SD = TRUE and model.based = FALSE

Details

If you have data in a matrix and information about experimental design factors, then you can use neweS to convert the data into an ExpressionSet object. Please see neweS for 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.

The default estimation method is maximum likelihood. The likelihood is derived by assuming that there exist values for lambda and alpha such that the residuals from the linear model in model, fit to glog-transformed data using those values for lambda and alpha, follow a normal distribution. See Durbin and Rocke (2003) for details.

If SD = TRUE, lambda and alpha are estimated by minimizing the stability score rather than by maximum likelihood. The stability score is defined as the absolute value of the slope coefficient from the regression of the replicate/residual standard deviation on the gene/row means, or on the

rank of the gene/row means. If `model.based = TRUE`, the stability score is calculated using the standard deviation of residuals from the linear model in `model`. Otherwise, the stability score is calculated using the pooled standard deviation over sets of replicates in `rep.arrays`. See Wu and Rocke (2009) for details.

Optimization methods in `method` are as follows:

1 = Newton-type method, using `nlm`

2 = Nelder-Mead, using `optim`

3 = BFGS, using `optim`

4 = Conjugate gradients, using `optim`

5 = Simulated annealing, using `optim` (may only be used when `mult = TRUE`)

Value

A list with components:

`lambda`	Estimate of transformation parameter lambda
`alpha`	Estimate of transformation parameter alpha

Author(s)

David Rocke, Geun-Cheol Lee, John Tillinghast, Blythe Durbin-Johnson, and Shiquan Wu

References

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

Wu, S. and Rocke, D.M. (2009) Analysis of Illumina BeadArray data using variance stabilizing transformations.

http://dmrocke.ucdavis.edu

See Also

`tranestAffyProbeLevel`, `lnorm`, `glog`

Examples

```
library(Biobase)
library(LMGene)

#data
data(sample.eS)

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

tranestAffyProbeLevel *Glog transformation parameter estimation function for probe-level Affymetrix expression data*

Description

Estimates parameters for the glog transformation on probe-level Affymetrix expression data, by maximum likelihood or by minimizing the stability score.

Usage

```
tranestAffyProbeLevel(eS, ngenes = 5000, starting = FALSE, lambda = 1000,
alpha = 0, gradtol = 0.001,lowessnorm = FALSE, method = 1, mult = FALSE,
model = NULL, SD = FALSE, rank = TRUE, model.based = TRUE,
rep.arrays = NULL)
```

Arguments

eS	An AffyBatch object
ngenes	Number of randomly sampled probesets to be used in estimating the transformation parameter
starting	If TRUE, user-specified starting values for lambda and alpha are input to the optimization routine
lambda	Starting value for parameter lambda. Ignored unless starting = TRUE
alpha	Starting value for parameter alpha. Ignored unless starting = TRUE
gradtol	A positive scalar giving the tolerance at which the scaled gradient is considered close enough to zero to terminate the algorithm
lowessnorm	If TRUE, lowess normalization (using lnorm) is used in calculating the likelihood.
method	Determines optimization method. Default is 1, which corresponds to a Newton-type method (see nlm and details.)
mult	If TRUE, tranest will use a vector alpha with one (possibly different) entry per sample. Default is to use same alpha for every sample. SD and mult may not both be TRUE.
model	Specifies model to be used. Default is to use all variables from eS without interactions. See details.
SD	If TRUE, transformation parameters are estimated by minimizing the stability score. See details.
rank	If TRUE, the stability score is calculated by regressing the replicate standard deviation on the rank of the probe/row means (rather than on the means themselves). Ignored unless SD = TRUE
model.based	If TRUE, the stability score is calculated using the standard deviation of residuals from the linear model in model. Ignored unless SD = TRUE
rep.arrays	List of sets of replicate arrays. Each element of rep.arrays should be a vector with entries corresponding to arrays (columns) in exprs(eS) conducted under the same experimental conditions, i.e., with identical rows in pData(eS). Ignored unless SD = TRUE and model.based = FALSE

Details

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.

The default estimation method is maximum likelihood. The likelihood is derived by assuming that there exist values for lambda and alpha such that the residuals from the linear model in model, fit to glog-transformed data using those values for lambda and alpha, follow a normal distribution. See Durbin and Rocke (2003) for details.

If SD = TRUE, lambda and alpha are estimated by minimizing the stability score rather than by maximum likelihood. The stability score is defined as the absolute value of the slope coefficient from the regression of the replicate/residual standard deviation on the probe/row means, or on the rank of the probe/row means. If model.based = TRUE, the stability score is calculated using the standard deviation of residuals from the linear model in model. Otherwise, the stability score is calculated using the pooled standard deviation over sets of replicates in rep.arrays. See Wu and Rocke (2009) for details.

A random sample of probsets (of size ngene) is sampled from featureNames(eS). Expression data from all probes in the sampled probesets is used in estimating the transformation parameters.

Optimization methods in method are as follows:

1 = Newton-type method, using nlm

2 = Nelder-Mead, using optim

3 = BFGS, using optim

4 = Conjugate gradients, using optim

5 = Simulated annealing, using optim (may only be used when mult = TRUE)

Value

A list with components:

lambda	Estimate of transformation parameter lambda
alpha	Estimate of transformation parameter alpha

Author(s)

Lei Zhou, David Rocke, Geun-Cheol Lee, John Tillinghast, Blythe Durbin-Johnson, and Shiquan Wu

References

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

Wu, S. and Rocke, D.M. (2009) Analysis of Illumina BeadArray data using variance stabilizing transformations.

Zhou, L. and Rocke, D.M. (2005) An expression index for Affymetrix GeneChips based on the generalized logarithm, *Bioinformatics*, **21**, 3983–3989.

http://dmrocke.ucdavis.edu

See Also

tranest, lnorm, psmeans, glog

Examples

```
library(LMGene)
library(affy)
library(Biobase)
library(affydata)

data(Dilution)

tranpar.Dilution <- tranestAffyProbeLevel(Dilution, model = "liver",
ngenes = 3000, method = 2)

# transform data
trans.Dilution <- transeS(Dilution, tranpar.Dilution$lambda,
tranpar.Dilution$alpha)

# extract transformed perfect matches
exprs(trans.Dilution) <- pm(trans.Dilution)

# lowess normalize transformed data
lnorm.Dilution <- lnormeS(trans.Dilution)
## Not run:
# Average over probesets
# First, create index of probes
fnames <- featureNames(Dilution)
p <- length(featureNames(Dilution))
ind <- vector()
for (i in 1:p){
nprobes <- dim(pm(Dilution,fnames[i]))[1]
ind <- c(ind, rep(i,nprobes))
}

avg.Dilution <- psmeans(lnorm.Dilution, ind)

## End(Not run)
```

`transeS` *Function to apply the glog transform to an expression set.*

Description

For each element in the array of expression data, this function applies the glog transform y -> glog (y-alpha, lambda). If alpha is a vector, it must have one element for each column in `exprs(eS)`.

Usage

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

Arguments

eS	An ExpressionSet or AffyBatch object
lambda	The parameter lambda to be used in the glog transform.
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.

Details

The glog transformation of a variable y is defined as `log(y + sqrt(y^2 + lambda))`. Using `lambda = 0` corresponds to the log transformation, up to a scale factor of 2. (Other, equivalent expressions exist for the glog transformation. See Durbin et al. (2002) and Huber et al. (2002) for futher details.)

`transeS` subtracts a (scalar or vector) parameter `alpha` prior to application of the glog transformation, resulting in the expression `log(y - alpha + sqrt((y - alpha)^2 + lambda))`.

The parameters `lambda` and `alpha` may be estimated using `tranest`.

Value

Returns an `ExpressionSet` or `AffyBatch` object with the expression matrix glog-transformed.

Author(s)

John Tillinghast

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.

Huber, W., Von Heydebreck, A., Sueltmann, H., Poustka, A., and Vingron, M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression, *Bioinformatics*, **18**, S96–S104.

http://dmrocke.ucdavis.edu

See Also

`glog`, `tranest`

Examples

```
library(LMGene)
library(Biobase)

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

vlist *Sample experimental/phenotype data for LMGene package*

Description

List of experimental factors for the sample matrix array data, 'sample.mat'.

Usage

```
data(vlist)
```

Examples

```
library(Biobase)
library(LMGene)

#data
data(vlist)

vlist
```

Index