Package ‘LMGene’

November 20, 2016

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 ................................................................. 2
glog ............................................................... 4
LMGene ............................................................. 5
lnorm ............................................................... 6
lnormeS ............................................................ 7
neweS ............................................................. 8
norm ................................................................. 9
plotMeanSD ....................................................... 10
psmeans ........................................................... 11
pvadjust ........................................................... 12
rowaov ............................................................ 13
sample.eS .......................................................... 14
sample.ind ........................................................ 15
sample.mat ........................................................ 16
tranest ............................................................. 16
tranestAffyProbeLevel ............................................ 19
transeS ............................................................ 21
vlist ............................................................... 23
**genediff**

**Index**

---

| 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**

```r
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


See Also

*LMGene, rowaov, neweS*

Examples

```r
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 further 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


http://dmrocke.ucdavis.edu

See Also

- tranest, tranes$
#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**

```r
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


http://dmrocke.ucdavis.edu

See Also

genediff, neweS

Examples

```r
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)
```

Description

Lowess normalization function

Usage

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

Arguments

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

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

Arguments

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

Coerce a matrix to class ExpressionSet

---

**Value**

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

**Author(s)**

John Tillinghast, Blythe Durbin-Johnson

**References**

[http://dmrocke.ucdavis.edu](http://dmrocke.ucdavis.edu)

**See Also**

`lnorm`, `norm`

**Examples**

```r
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)
```

---

**Description**

This function converts a data matrix into an ExpressionSet object.

**Usage**

```r
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.
**norm**

**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))

---

<table>
<thead>
<tr>
<th><strong>norm</strong></th>
<th><strong>Normalization function</strong></th>
</tr>
</thead>
</table>

**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
plotMeanSD

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 auto-
matically 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
**psmeans**

**Author(s)**

Rachel Chen and Blythe Durbin-Johnson

**Examples**

```r
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**

```r
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

```r
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

```r
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/
rowaov

See Also
genediff

Examples

```r
#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**

```r
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]
Details

Identical with neweS(sample.mat, vlist), up to metadata

Examples

```r
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 probeset index vector**

---

Description

Vector indicating which probeset each probe belongs to

Usage

```r
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

```r
data(sample.eS)
data(sample.ind)
trs.eS <- transeS(sample.eS, 667, 65)
ntres.eS <- lnormeS(trs.eS)
genessample.eS <- psmeans(ntrs.eS, sample.ind)
```
**sample.mat**  
*Sample array data for LMGene package*

**Description**

A matrix of array data

**Usage**

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

**Format**

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

**Examples**

```r
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**

```r
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 \texttt{model.based = TRUE}, the stability score is calculated using the standard deviation of residuals from the linear model in \texttt{model}. Otherwise, the stability score is calculated using the pooled standard deviation over sets of replicates in \texttt{rep.arrays}. See Wu and Rocke (2009) for details.

Optimization methods in \texttt{method} are as follows:

1 = Newton-type method, using \texttt{nlm}
2 = Nelder-Mead, using \texttt{optim}
3 = BFGS, using \texttt{optim}
4 = Conjugate gradients, using \texttt{optim}
5 = Simulated annealing, using \texttt{optim} (may only be used when \texttt{mult = TRUE})

**Value**

A list with components:

- \texttt{lambda} Estimate of transformation parameter lambda
- \texttt{alpha} Estimate of transformation parameter alpha

**Author(s)**

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

**References**


[http://dmrocke.ucdavis.edu](http://dmrocke.ucdavis.edu)

**See Also**

\texttt{tranestAffyProbeLevel}, \texttt{lnorm}, \texttt{glog}

**Examples**

```r
library(Biobase)
library(LMGene)

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

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

**Usage**

```r
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 `model1`. 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 probsets 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


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 \( y \rightarrow \text{glog}(y - \alpha, \lambda) \). If \( \alpha \) is a vector, it must have one element for each column in exprs(eS).

**Usage**

transeS(eS, lambda, alpha)
transeS

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 further 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


http://dmrocke.ucdavis.edu

See Also

glog, tranest

Examples

```r
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

*Topic **datasets**
  - sample.eS, 14
  - sample.ind, 15
  - sample.mat, 16
  - vlist, 23

*Topic **dplot**
  - plotMeanSD, 10

*Topic **manip**
  - neweS, 8
  - psmmeans, 11

*Topic **math**
  - glog, 4
  - tranest, 16
  - tranestAffyProbeLevel, 19
  - traneS, 21

*Topic **models**
  - genediff, 2
  - pvadjust, 12
  - rowaov, 13

*Topic **smooth**
  - lnorm, 6
  - lnormeS, 7
  - genediff, 2, 5, 6, 13, 14
  - glog, 4, 18, 21, 22

LMGene, 2, 3, 5, 13, 14
lnorm, 6, 7, 8, 10, 17–19, 21
lnormeS, 7, 7
neweS, 2, 3, 5, 6, 8, 13, 17
norm, 7, 8, 9
plotMeanSD, 10
psmeans, 11, 21
pvadjust, 12
rowaov, 3, 13
sample.eS, 14
sample.ind, 11, 15
sample.mat, 16

tranest, 4, 16, 21, 22
tranestAffyProbeLevel, 11, 18, 19