Package 'anapuce'

October 12, 2022

Title Tools for Microarray Data Analysis

Version 2.3

Description Functions for normalisation, differentially analysis of microarray data and local False Discovery Rate.

Depends R (>= 3.4.0)

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Julie Aubert [aut, cre]

Maintainer Julie Aubert <julie.aubert@agroparistech.fr>

Repository CRAN

Date/Publication 2019-02-05 15:13:20 UTC

R topics documented:

apo *Apo AI data*

Description

An exemple data set for unpaired data analysis. A dataframe with the normalized Apo data set as used in the original article.

Format

apo is a dataframe. The first column contains gene names. The 8 columns cond1.1 to cond1.8 contain normalized normal mice measurements and the 8 columns cond2.1 to cond2.8 correspond to normalized KO mice measurements

References

M.J. Callow, S. Dudoit, E.L. Gong, T.P. Speed, and E.M. Rubin. Microarray expression profiling identifies genes with altered expression in hdl-deficien mice. Genome Res., 10(12) : 2022-9, 2000

See Also

DiffAnalysis.unpaired

Examples

```
data(apo)
# resApo <- DiffAnalysis.unpaired(apo,n=1,ind.array=1:8,varmixt.meth=TRUE)
# Histogramm of the raw-pvalues
# hist(resApo$PValueVM)
```

DiffAnalysis *Differential Analysis for paired data*

Description

Performs the differential analysis from normalised paired microarray data according to different ways of variance modelling and computes lists of differentially expressed genes according different multiple testing procedures.

Usage

```
DiffAnalysis(fileIN = "resNorm.txt", n = 3, ind.array = 1:2, name.A = "A",
name.M = "M.norm", fileOUT = "ListOfGenes.txt",
fileDelete = "GenesOutOfAnalysis.txt", procs = c("bonferroni", "BH"),
alpha = c(0.05, 0.05), dyeswap = TRUE, indDS = c(2), fileID = NULL,
function.trt = NULL, by.var = "ID", varmixt.meth = TRUE, header = TRUE,
sep = "\t", sep.write = "\t", dec.write = ".", ...)
```

Arguments

fileIN	normalized data set.
n	number of identificant columns.
ind.array	the indices of arrays to analyze.
name.A	character string containing a regular expression contained in the columnames corresponding to the A values.
name.M	character string containing a regular expression contained in the columnames corresponding to the M values.
fileOUT	output data file.
fileDelete	output data file containing the list of withdrawn genes.
procs	adjusting p-values procedures to be used.
alpha	values of the first type error to be used in the different procedures.
dyeswap	logical indicating whether the design is a dye swap.
indDS	index of arrays related to dye swap.
fileID	file giving information about genes.
function.trt	function to be applied before differential analysis.
by.var	argument passed to function.trt.
varmixt.meth	logical indicating whether to perform the variance modelling proposed by Delmar et al. (2005).
header	a logical value indicating whether the file contains the names of the variables as its first line. Used in the read.table function.
sep	the field separator string to use in the read.table function.
sep.write	the field separator string to use in the write.table function.
dec.write	the string to use for decimal points in the write.table function.
...	Further arguments to be passed to read.table.

Details

This function performs a differential analysis in the gene-specific and homoscedastic cases. If *varmixt.meth* = *TRUE*, the method used is the method proposed by Delmar et al. (2005), (the same as coded in the vm.analysis.paired function from the varmixt package.)

Value

By default these following files are created

- one list of differentially expressed genes by procedure

- The list of withdrawn genes

- The list of all the genes

An R object is returned if function return value is assigned.

Author(s)

J. Aubert

References

Delmar, P., Robin, S. and Daudin, J.J., (2005), VarMixt: efficient variance modelling for the differential analysis of replicated gene expression data, *Bioinformatics*, **21**,(4), 502–8

Dudoit, S., Yang, Y. H., Callow, M. J. and Speed, T.P., (2002), Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments, *Statisticia Sinica*, **12**, 111–139

See Also

`p.adjust`, `MeanBySpot`, `DiffAnalysis.unpaired`

Examples

```
data(spleen)
# Analysis on the first 100 genes
resSpleen <- DiffAnalysis(spleen[1:100, ], n = 1, ind.array = 1:6, name.A = "A.", name.M = "M.",
varmixt.meth = TRUE, dyeswap = TRUE, indDS = c(2,4,6))
# Histogramm of the raw-pvalues
# hist(resSpleen$PValueVM)
```

DiffAnalysis.unpaired *Differential Analysis for unpaired data*

Description

Performs the differential analysis from normalised unpaired data according to different ways of variance modelling and computes lists of differentially expressed genes according different multiple testing procedures.

Usage

```
DiffAnalysis.unpaired(fileIN = "resNorm.txt", n = 3, cond1 = "cond1.",
cond2 = "cond2.", fileOUT = "ListOfGenes.txt",
fileDelete = "GenesOutOfAnalysis.txt", procs = c("bonferroni", "BH"),
alpha = c(0.05, 0.05), fileID = NULL, function.trt = NULL,
by.var = "ID", varmixt.meth = TRUE, header = TRUE, sep = "\t",
sep.write = "\t", dec.write = ".", ...)
```

Arguments

fileIN	normalized data set.
n	number of identificant columns.
cond1	a regular expression corresponding to the first condition.
cond2	a regular expression corresponding to the second condition.
fileOUT	output data file.
fileDelete	output data file containing the list of withdrawn genes.
procs	adjusting p-values procedures to be used.
alpha	values of the first type error to be used in the different procedures.
fileID	file giving information about genes.
function.trt	function to be applied before differential analysis.
by.var	argument passed to function.trt.
varmixt.meth	logical indicating whether to perform the variance modelling proposed by Delmar et al. (2005).
header	a logical value indicating whether the file contains the names of the variables as its first line. Used in the read.table function.
sep	the field separator string to use in the read.table function.
sep.write	the field separator string to use in the write.table function.
dec.write	the string to use for decimal points in the write.table function.
...	Further arguments to be passed to read.table.

Details

This function performs a differential analysis in the gene-specific and homoscedastic cases.

If *varmixt.meth = TRUE*, the method used is the method proposed by Delmar et al. (2005), (the same as coded in the vm.analysis function from the varmixt package.)

Value

By default these following files are created

- one list of differentially expressed genes by procedure

- The list of withdrawn genes

- The list of all the genes

An R object is returned if function return value is assigned.

Author(s)

J. Aubert

References

Delmar, P., Robin, S. and Daudin, J.J., (2005), VarMixt: efficient variance modelling for the differential analysis of replicated gene expression data, *Bioinformatics*, **21**,(4), 502–8

Dudoit, S., Yang, Y. H., Callow, M. J. and Speed, T.P., (2002), Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments, *Statisticia Sinica*, **12**, 111–139

See Also

`p.adjust`, `MeanBySpot`, `DiffAnalysis`

Examples

```
data(apo)
# Analysis on the first 100 genes
resApo <- DiffAnalysis.unpaired(apo[1:100,], n = 1, ind.array = 1:8, varmixt.meth = TRUE)
# Histogramm of the raw-pvalues
# hist(resApo$PValueVM)
```

---

est.varmixt *Variance Mixture Estimation*

Description

Performs variance mixture analysis

Usage

```
est.varmixt(VAR, Kmax, dfreedom)
```

Arguments

`VAR`	vector of estimated variance.
`Kmax`	maximal number of variance components.
`dfreedom`	degrees of freedom of the estimated variance.

Value

a LIST with the following components :

`BIC.crit`	value of the BIC criterion
`p.i`	the probability of each variance component
`vars`	variances
`loglike`	value of the criterion based on Loglikelihood
`nmixt`	number of variance components

tau	the matrix of posterior probability that a gene belongs to each variance component. One row per gene, one column per variance component.
VM2	the variance attributed to each gene according to the MAP rule - Delmar et al. (2005) *JRSS*
VM	the variance attributed to each gene taking into account the tau values - Delmar et al. (2005) *Bioinformatics*

Author(s)

M-L Martin-Magniette and J. Aubert

References

Delmar P, Robin S, Le Roux D, Daudin J.J (2005), Mixture model on the variance for the differential analysis of gene expression, *JRSS series C*, **54:1**, 31-50.

Delmar P, Robin S, Daudin J.J (2005), VarMixt: efficient variance modelling for the differential analysis of replicated gene expression data, *Bioinformatics*, **21(4)**, 502-8.

See Also

DiffAnalysis,DiffAnalysis.unpaired

---

fdr.estimate.eta0 *Estimate the Proportion of Null p-Values*

Description

fdr.estimate.eta0 estimates the proportion eta0 of null p-values in a given vector of p-values. This quantity is an important parameter when controlling the false discovery rate (FDR). A conservative choice is eta0 = 1 but a choice closer to the true value will increase efficiency and power - see Benjamini and Hochberg (1995, 2000) and Storey (2002) for details.

Usage

```
fdr.estimate.eta0(p, method=c("conservative", "adaptive", "bootstrap",
 "smoother"), lambda=seq(0,0.95,0.05) )
```

Arguments

p	vector of p-values.
method	algorithm used to estimate the proportion of null p-values. Available options are "conservative" (default), "adaptive", "bootstrap", and "smoother".
lambda	optional tuning parameter vector needed for "bootstrap" and "smoothing" methods (defaults to seq(0,0.95,0.05)) - see Storey (2002) and Storey and Tibshirani (2003).

Details

The function `fdr.estimate.eta0` provides four algorithms: the "conservative" method always returns eta0 = 1 (Benjamini and Hochberg, 1995), "adaptive" uses the approach suggested in Benjamini and Hochberg (2000), "bootstrap" employs the method from Storey (2002), and "smoother" uses the smoothing spline approach in Storey and Tibshirani (2003).

Value

The estimated proportion eta0 of null p-values.

Author(s)

Konstantinos Fokianos and Korbinian Strimmer.

Adapted in part from S-PLUS code by Y. Benjamini and R code from J.D. Storey (http://genomics.princeton.edu/storeylab/).

References

"conservative" procedure: Benjamini, Y., and Y. Hochberg (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. B*, **57**, 289–300.

"adaptive" procedure: Benjamini, Y., and Y. Hochberg (2000) The adaptive control of the false discovery rate in multiple hypotheses testing with independent statistics. *J. Behav. Educ. Statist.*, **25**, 60–83.

"bootstrap" procedure: Storey, J. D. (2002) A direct approach to false discovery rates. *J. Roy. Statist. Soc. B.*, **64**, 479–498.

"bootstrap" procedure: Storey, J. D., and R. Tibshirani (2003) Statistical significance for genome-wide experiments. *Proc. Nat. Acad. Sci. USA*, **100**, 9440-9445.

`filterByDefault` *Filter applied on data before normalisation*

Description

Apply a filter on input data by deleting spots with values for filter.var are mentionned in *filter.fic* by deleting spots with flags equal to *flag0* (-100 by default).

Usage

```
filterByDefault(fileIN, flag0, m, filter.fic = filter.fic,
filter.var = filter.var, sep = sep, sep.write = sep.write,
dec.write = dec.write, ...)
```

Arguments

fileIN	input datafame.
flag0	value(s) of flags to delete.
m	number of identificant columns.
filter.fic	name of the file containing the values of the variable named filter.var to delete.
filter.var	name of the variable concerned with deletion.
sep	the field separator string to use in the read.table function.
sep.write	the field separator string to use in the write.table function.
dec.write	the string to use for decimal points in the write.table function.
...	Further arguments to be passed to read.table.

Value

A subset of the input dataframe

Author(s)

J. Aubert

See Also

normalisation

Intensity.Norm *Function to compute normalized intensity for each channel*

Description

Computes normalized intensity for each channel and each array from the file containing normalized data.

Usage

```
Intensity.Norm(fileIN = "resNorm.txt", n = 3, ind.array = NULL,
name.A = "A", name.M = "M.norm", sep = "\t", center = FALSE,
log.transf = TRUE, ...)
```

Arguments

fileIN	input dataframe.
n	number of identificant columns.
ind.array	indices of arrays to use.
name.A	character string containing a regular expression contained in the columnames corresponding to the A values.

name.M	character string containing a regular expression contained in the columnames corresponding to the M values.
sep	the field separator string to use in the read.table function.
center	logical indicating wether data will be centered by array.
log.transf	logical indicating wether data will stay log-transformed.
...	Further arguments to be passed to read.table.

Value

A R dataframe containing the n first columns of the input dataframe and the "normalized" intensity for the red and green channels of each chosen array.

Author(s)

J. Aubert

References

Thorne, N.P. (2004) Single-channel normalisation and analysis of twocolour cDNA microarray data, PhD thesis

See Also

normalisation

LBI *Label Bias Index*

Description

This function calculates the LBI (Label Bias Index) for two self-self hybridization microarrays.

Usage

```
LBI(infile, name.M = "M.norm", ind.array = 1:2, graph = TRUE,
graphout = "FigM1M2")
```

Arguments

infile	input data file with normalized log-ratio base 2.
name.M	character string containing a regular expression contained in the columnames corresponding to the M values.
ind.array	the indices of arrays to analyze.
graph	logical indicating wether to perform graphics.
graphout	name of the graphic output.

Author(s)

J. Aubert

References

Martin-Magniette M.L, Aubert J, Cabannes E, Daudin J.J (2005) Evaluation of the gene-specific dye bias in cDNA microarray experiments, *Bioinformatics*, **21(9)**, 1995-2000.

LocalFDR *Local FDR*

Description

Estimation of local false discovery rates

Usage

```
LocalFDR(dataf = dataf, graph = TRUE, method = NULL, lambda0 = 0.5,
smoothing = "1", thres = c(0.01, 0.05, 0.1, 0.2), mm = c(3, 5, 15, NA))
```

Arguments

dataf	input data file with two columns (1 = gene name, 2= ordered raw pvalues).
graph	if TRUE a pdf graphic is created.
method	method for estimating m0. This must be one of the strings "adaptive", "conservative", "bootstap", "smoother" or NULL.
lambda0	value used in calculating m0.
smoothing	"1" for the initial published method, "2" for the PAVA method isotonic (monotonely increasing nonparametric) least squares regression - see P. Broberg (2005).
thres	threshold defining intervals used for the movering average smoothing.
mm	parameter defining intervals used for the movering average smoothing.

Value

- if graph = TRUE, a file of graphics named LocalFDRGraph.pdf

- A data file (LocalFDRFile.txt) with 4 columns : gene name, raw pvalues and two columns corresponding to smoothed FDR values.

Author(s)

J. Aubert

References

Aubert J, Bar-Hen A, Daudin J.J, Robin S (2004) Determination of the differentially expressed genes in microarrays experiments using local FDR, *BMC Bioinformatics*, **5:125**.

Aubert J , Bar-Hen A, Daudin J.J, Robin S (2005) Correction: Determination of the differentially expressed genes in microarray experiments using local FDR, *BMC Bioinformatics*, **6:42**.

Per Broberg (2005) A comparative review of estimates of the proportion unchanged genes and the false discovery rate, *BMC Bioinformatics* **6:199**

`MeanBySpot` *Mean By Spot*

Description

Computes the mean on the different values of spots grouping by *by.var*.

Usage

```
MeanBySpot(fileIN, n = 3, name.A = "A", name.M = "M.norm",
by.var = "ID", na.rm=TRUE)
```

Arguments

`fileIN`	input dataframe.
`n`	number of identificant columns.
`name.A`	character string containing a regular expression contained in the columnames corresponding to the A values.
`name.M`	character string containing a regular expression contained in the columnames corresponding to the M values.
`by.var`	name of the grouping variable.
`na.rm`	a logical value indicating whether NA values should be stripped before the computation proceeds

Value

A R dataframe

Author(s)

J. Aubert

See Also

`DiffAnalysis`, `DiffAnalysis.unpaired`,`mean`

`normalisation` *Function to normalize microarray data*

Description

Performs the data normalization using a general loess and a block effect correction via the subtraction of the median by block if there is a Block column.

Usage

```
normalisation(fileIN = NULL, Red = "F635.Median", Green = "F532.Median",
n = 3, flag = -100, graph = TRUE, filter.function = filterByDefault,
filter.fic = NULL, filter.var = NULL, sep.write = "\t", dec.write = ".",
header = TRUE, sep = "\t", skip = 0, ...)
```

Arguments

`fileIN`	on optional regular expression contained in names of the files to analyze
`Red`	name of the column containing the Cy5 values
`Green`	name of the column containing the Cy3 values
`n`	number of identificant columns
`flag`	list of flags values to delete
`graph`	logical indicating wether to perform graphics
`filter.function`	function to perform before beginning the normalization
`filter.fic`	argument passed to filter.funtion - file containing the values of the variable named *filter.var* to delete
`filter.var`	argument passed to filter.funtion
`sep.write`	the field separator string to use in the write.table function
`dec.write`	the string to use for decimal points in the write.table function
`header`	a logical value indicating whether the file contains the names of the variables as its first line. Used in the read.table function
`sep`	the field separator string to use in the read.table function
`skip`	integer: the number of lines of the data file to skip before beginning to read data.
`...`	Further arguments to be passed to read.table.

Details

By default *fileIN=NULL* and the files to analyze have to be choosen among all the files in the working directory. If fileIN is not NULL, only file names which match the regular expression fileIN will be proposed to the normalization step.

Value

A file *resNorm.txt* containing the normalized data and if *graph=TRUE* a file .pdf per array with graphics

Author(s)

J. Aubert

References

Yang, Y., Dudoit, S., Luu, P., Lin, D., Peng, V., Ngai, J. and Speed, T., (2002), Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation, *Nucleic Acids Research* **30**.

See Also

`filterByDefault`,`read.table`

`spleen` *Spleen Data set*

Description

An example data set for paired data analysis. A dataframe with the normalized Spleen data set as used in the original article.

Format

spleen is a dataframe. The first column contains gene names. The 6 columns A.1 to A.6 contain normalized mean log-intensity and the 6 columns M.1 to M.6 correspond to normalized log-ratio

References

P. Delmar, Robin, S., Tronik-Le Roux S. and Daudin J.-J. (2005) Mixture model on the variance for the differential analysis of gene expression data, JRSS series C, 54(1), 31:50

See Also

`DiffAnalysis`

Examples

```
data(spleen)
# resSpleen <- DiffAnalysis(spleen,n=1,ind.array=1:6,name.A="A.",name.M="M.",
# varmixt.meth=TRUE,dyeswap=TRUE,indDS=c(2,4,6))
# Histogramm of the raw-pvalues
# hist(resSpleen$PValueVM)
```

Index