Package: fcros (via r-universe)

September 8, 2024

Version 1.6.1

Date 2019-05-28

Title A Method to Search for Differentially Expressed Genes and to Detect Recurrent Chromosomal Copy Number Aberrations

Author Doulaye Dembele

Maintainer Doulaye Dembele <doulaye@igbmc.fr>

Depends R (>= 3.1.0)

Description A fold change rank based method is presented to search for genes with changing expression and to detect recurrent chromosomal copy number aberrations. This method may be useful for high-throughput biological data (micro-array, sequencing, ...). Probabilities are associated with genes or probes in the data set and there is no problem of multiple tests when using this method. For array-based comparative genomic hybridization data, segmentation results are obtained by merging the significant probes detected.

License GPL (>= 2)

NeedsCompilation yes

Date/Publication 2019-05-31 12:40:07 UTC

Repository https://doulaye.r-universe.dev

RemoteUrl https://github.com/cran/fcros

RemoteRef HEAD

RemoteSha 80a62f508adde2f33168ba4bb9d86c1f7e168dcb

Contents

`fcros-package` *A Method to Search for Differentially Expressed Genes and to Detect Recurrent Chromosomal Copy Number Aberrations*

Description

Implementation of a method based on fold change rank ordering statistics to search for differentially expressed genes or to detect recurrent chromosomal copy number aberrations. This package can be used for two biological conditions high-throughput dataset (microarray, RNA-seq, ...), for expression profiling dataset over time without replicates or for cytogenetics dataset (aCGH, Sequencing).

Details

Package:	fcros
Type:	Package
Version:	1.6.1
Date:	2019-05-28
License:	GPL (>= 2)

Package fcros has the following functions:

fcros():	The function to use with a dataset from two biological condition samples. The dataset should be in a single table. The function fcros() performs a pairwise conparison of samples to obtain a matrix of fold changes. The fold changes are sorted and their rank values are used to associate statistic with genes/probes.
fcros2():	The function to use with datasets from two biological biological conditions. The datasets should be in two separate tables as inputs. The function fcros2() performs a pairwise comparison of samples from each table to obtain fold changes. The fold changes are sorted, their rank values are combined and then used to associate statistic with genes/probes.
pfco():	The function to use with a dataset from two biological condition samples. The dataset should be in a single table. The function pfco() performs a pairwise conparison of samples to obtain a matrix of fold changes. The fold changes are sorted and their rank values are used to associate statistic with genes/probes using a singular value decomposition.
fcrosMod():	This function uses fold changes or ratios matrix as input to associate statistic with genes/probes.
pfcoMod():	This function uses fold changes or ratios matrix as input to associate statistic with genes/probes using a singular value decomposition.
fcrosFCmat():	This function allows to compute a matrix of fold changes using pairwise comparisons of the two biological condition samples in a dataset.
fcrosTtest():	This function allows to use the Student t-test to calculate p-values for the genes in a dataset.
fcrosRead():	This function allows to read a tab delimited text file to be use as an input for the function fcros(), fcros2() or fcrosMod().
fcrosWrite():	This function allows to save the results obtained using the function fcros(), fcros2() or fcrosMod() in a tab delimited text file.
fcrosTopN():	This function allows to search for the top N down- and/or up-regulated genes from the results obtained using the function fcros(), fcros2(), pfco(), fcrosMod() or pfcoMod().
fvalTopN():	This function allows to search for the top N down- and/or up-regulated genes from the results obtained using the function fcros(), fcros2(), pfco(), fcrosMod() or pfcoMod().
pvalTopN():	This function allows to search for the top N down- and/or up-regulated genes from the results obtained using the function fcros(), fcros2(), pfco(), fcrosMod() or pfcoMod().
histoPlot():	This function plots on the screen the histogram of the FCROS statistics obtained using the results of the function fcros(), fcros2(), pfco(),

	fcrosMod() or pfcoMod()
fvalVolcanoPlot():	This function performs a volcano plot of the results obtained using the function fcros(), fcros2(), pfco(), fcrosMod() or pfcomod()
pvalVolcanoPlot():	This function performs a volcano plot of the results obtained using the function fcros(), fcros2(), pfco(), fcrosMod() or pfcoMod()
chrSummary():	This function summarizes detection results by chromosome
chrSegment():	This function segments a chromosome data
chrPlot():	This function performs a plot of the chromosome probes data
chrPlot2():	This function performs a plot of the chromosome segmentation results
voomReads():	This function performs a transformation of the read counts
tcnReads():	This function performs a total count normalization of reads
rankReads():	The function to use with a dataset from two biological condition samples. The dataset should be in a single table. The function rankReads() performs a pairwise conparison of samples to obtain a matrix of fold changes. Small uniform values are added to read counts. This is repeated nrun time. The fold changes are sorted and their rank values are used to associate statistic with genes/probes.
scoreThr():	Using the log10 transformed score values obtained with the rankReads(), this function computes numerically the inflection point value given lower and upper bound values for the slope region.

Author(s)

Doulaye Dembele Maintainer: Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
trim.opt <- 0.25;

# perform fcros()
af <- fcros(fdata, cont, test, log2.opt, trim.opt);
```

```
# perform Volcano plot
fvalVolcanoPlot(af, thr = 0.01)

# save fcros values in a file
fcrosWrite(af, file = "test2delete_values.txt");

# now select top 20 down and/or up regulated genes
top20 <- fcrosTopN(af, 20);
alpha1 <- top20$alpha[1];
alpha2 <- top20$alpha[2];
id.down  <- matrix(c(0,11), ncol = 1);
id.up <- matrix(c(rep(0,11)), ncol = 1);
n <- length(af$FC);
f.value <- af$f.value;

idown <- 1;
iup <- 1;
for (i in 1:n) {
    if (f.value[i] <= alpha1) { id.down[idown] <- i; idown <- idown+1; }
    if (f.value[i] >= alpha2) { id.up[iup] <- i; iup <- iup+1; }
}

data.down <- fdata[id.down[1:(idown-1)], ];
ndown <- nrow(data.down);
data.up <- fdata[id.up[1:(iup-1)], ];
nup <- nrow(data.up);

# now plot down regulated genes
t <- 1:20;
op = par(mfrow = c(2,1));
plot(t, data.down[1,2:21], type = "l", col = "blue", xlim = c(1,20),
       ylim = c(0,18), main = "Top down-regulated genes");
for (i in 2:ndown) {
    lines(t, data.down[i,2:21], type = "l", col = "blue")
}

# now plot down and up regulated genes
plot(t, data.up[1,2:21], type = "l", col = "red", xlim = c(1,20),
    ylim = c(0,18), main = "Top up-regulated genes");
for (i in 2:nup) {
    lines(t, data.up[i,2:21], type = "l", col = "red")
}
par(op)
```

---

bott *Example of sequencing data to test the rankReads function.*

Description

This is a subset of data taken from the Bottomly dataset see http://bowtie-bio.sourceforge.net/recount/. The complete dataset has 36,536 rows or ENSEMBL identifiers (genes) and 21 columns (samples).

For the "bott" data, the first 5,000 rows, first 3 and last 3 samples were used.

Usage

```
data(bott)
```

Format

A data frame with 5,000 rows and 7 columns.

`gene`: ENSEMBL ID

`SRX033480`: sequencing values for the first B6 mouse

`SRX033488`: sequencing values for the second B6 mouse

`SRX033481`: sequencing values for the third B6 mouse

`SRX033493`: sequencing values for the first D2 mouse

`SRX033486`: sequencing values for the second D2 mouse

`SRX033494`: sequencing values for the third B6 mouse

Details

"bott" is a subset of a complet dataset obtained using a single RNA-seq reads from C57BL/6J (B6) and DBA/J2 (D2) mice.

References

Bottomly et al. Evaluating Gene Expression in C57BL/6J and DBA/J2 Mouse Striatum Using RNA-seq and Microarrays, PLoS One, 6(3)e17820, 2011

Examples

```
data(bott)
summary(bott)
```

---

calcSRmat *Calculation of the sorted rank matrix from the dataset*

Description

This is an internal function used to calculate the sorted rank matrix. It is used in the functions: fcros() and pfco().

Usage

```
calcSRmat(xdata, cont, test, log2.opt=0, trim.opt=0.25)
```

Arguments

xdata	A matrix or a table containing two biological conditions dataset to process for detecting differentially expressed genes: xdata.
cont	A vector containing the label names of the control samples: cont = c("cont01", "cont02", ...).
test	A vector containing the label names of the test samples: test = c("test01", "test02", "test03", ...).
log2.opt	A scalar equals to 0 or 1. The value 0 (default) means that data in the matrix "xdata" are expressed in a log2 scale: log2.opt = 0
trim.opt	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values of each gene are not used for computing its statistics "ri", i.e. the interquartile range rank values are averaged: trim.opt = 0.25

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kaster P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC bioinformatics, 2014, 15:14

Examples

```
#    data(fdata);
```

---

calcSRmatMod *Calculation of the sorted rank matrix from the dataset*

Description

This is an internal function used to calculate the sorted rank matrix. It is used in the functions: fcrosMod() and pfcoMod().

Usage

```
calcSRmatMod(xdata, samp, log2.opt=0, trim.opt=0.25)
```

Arguments

xdata	A matrix containing fold changes or ratios from a biological dataset to process for searching differentially expressed genes or for detecting recurrent copy number aberrations regions: fcMat.
samp	A vector of sample label names which should appear in the columns of the matrix fcMat: samp.

`log2.opt` A scalar equals to 0 or 1. The value 0 (default) means that values in the matrix "fcMat" are expressed in a log2 scale: `log2.opt = 0`

`trim.opt` A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values for each gene are not used for computing the statistic "ri", i.e. the interquartile range rank values are averaged: `trim.opt = 0.25`

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kaster P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC bioinformatics, 2014, 15:14

Examples

```
#    data(fdata);
```

---

cghData *Example of aCGH "data file" for the fcros package.*

Description

This dataset is part of the study performed and published by Sircoulomb et al. 2010, BMC Cancer, 10:539. For our illustration, we used the first 10 patients' data and for only 3 chromosomes: 7, 8 and 9. The complete data are available from the Gene Expression Omnibus website under the accession number GSE17907.

Usage

```
data(cghData)
```

Format

A data frame with 33,3613 observations for 10 samples.

`Probes`**:** a text for the unique probe ID

`GSM447252`**:** a numeric vector with log2 change values

`GSM447253`**:** a numeric vector with log2 change values

`GSM447254`**:** a numeric vector with log2 change values

`GSM447255`**:** a numeric vector with log2 change values

`GSM447256`**:** a numeric vector with log2 change values

`GSM447257`**:** a numeric vector with log2 change values

`GSM447258`**:** a numeric vector with log2 change values

`GSM447259`: a numeric vector with log2 change values

`GSM447260`: a numeric vector with log2 change values

`GSM447261`: a numeric vector with log2 change values

Details

This is a subset of a complete dataset obtained using array Comparative Genomic Hybridization technology. Agilent 244K design arrays have been used to monitor breast cancer patients.

References

F Sircoulomb, I Bekhouche, P Finetti, J Adelaide, AB Hamida, J Bonansea, S Raynaud, C Innocenti, E Charafe-Jauffret, C Tarpin, FB Ayed, P Viens, J Jacquemier, F Bertucci, D Birnbaum and M Chaffanet; Genome profiling of ERBB2-amplifief breast cancers. BMC Cancer, 2010, 10:539

Examples

```
data(cghData)

summary(cghData)

#  perform boxplot of data sample values
boxplot(cghData[,2:11])
```

---

cghInfo *Example of aCGH "info file" for the fcros package.*

Description

This dataset is part of the Agilent 244K design array probes description file which is used in the publicatin of Sircoulomb et al. 2010, BMC Cancer, 10:539. For our illustration, we used only 3 chromosomes (7, 8 and 9) data. The complete data are available in the file "GPL9158-3352.txt", see the Gene Expression Omnibus website and accession number GSE17907.

Usage

```
data(cghInfo)
```

Format

A data frame with 33,3613 rows for 7 columns with information on probes.

`Index`: a numeric used for the probe

`ProbeName`: a text for the unique probe ID

`GeneSymbol`: a text with the gene symbol associated with the probe

`Chromosome`: a text with the chromosome index associated with the probe

`Start`: a start position value for the sequence associated with the probe

`End`: an end position value for the sequence associated with the probe

`Cytoband`: a text for the cytoband associated with the probe

Details

This is a part of information obtained from the Agilent 244K design array probes description file.

References

F Sircoulomb, I Bekhouche, P Finetti, J Adelaide, AB Hamida, J Bonansea, S Raynaud, C Innocenti, E Charafe-Jauffret, C Tarpin, FB Ayed, P Viens, J Jacquemier, F Bertucci, D Birnbaum and M Chaffanet; Genome profiling of ERBB2-amplifief breast cancers. BMC Cancer, 2010, 10:539

Examples

```
data(cghInfo)

summary(cghInfo)
```

---

chrMerge *Using a C code for merging chromosome segments*

Description

This is an internal function for using a C code while merging chromosome segments in the segmentation step.

Usage

```
chrMerge(nbSeg, idStart, idEnd, lBound, uBound, segVal, segProba,
                        fcall, L2R, nd, dm, sigma)
```

Arguments

nbSeg	Number of current segments
idStart	Position indexes of the first probes for segments
idEnd	Positions indexes of the last probes for segments
lBound	Lower bound position for segments
uBound	Upper position for segments
segVal	Change values associated with segments
segProba	Probabilities associated with segments
fcall	Detection status associated with probes
L2R	Change values associated with probes
nd	Number of acceptable non-detection between two significant of a segment
dm	Average distance between two consecutive probes of the chromosome
sigma	Standard deviation of the residual observations, see reference

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
#    data(fdata);
```

---

chrPlot *Plot a chromosome data*

Description

This function generates a picture using a chromosome data.

Usage

```
chrPlot(chrData, thr = 0.05, deb = 100, fin = 1e10)
```

Arguments

chrData	A chromosome data obtained from an output of the function dataSummary(): xinfo2 = dataSummary(af, xinfo, chromosomes, alpha) idx = which(xinfo2$xinfo.s$Chromosome == "chr1") chrData = xinfo2$xinfo.s[idx,]
thr	The probability threshold leading to the selection of the significant probes: thr = 0.05
deb	This parameter allows to specify the start position of the chromosome region for plotting. It can be used for zooming. Negative value will lead to the plot of all chromosome data. deb = 100
fin	This parameter allows to specify the end position of the chromosome region for plot. It can be used for zooming. Negative value will lead to the plot of all chromosome data. thr = 1e7

Value

This function generates a picture on the screen

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
# load CGH data and info files
data(cghData)
rownames(cghData) <- cghData[,1];
data(cghInfo)
noms <- colnames(cghData)
m <- length(noms)
samp  <- noms[2:m]

# associate statistics with probes
af <- fcrosMod(cghData, samp, log2.opt = 0, trim.opt = 0.25)

chromosomes = c(7:9)
alpha <- 0.05

# summarize results for each chromosome
xinfo2 <- dataSummary(af, cghInfo, chromosomes, alpha)

# focused on chromosome 7 data
idx <- which(xinfo2$xinfo.s$Chromosome == "7")
chrData <- xinfo2$xinfo.s[idx, ]

# Plot chromosome 7 data
chrPlot(chrData, thr = alpha)
```

---

chrPlot2 *Plot a chromosome segmentation results*

Description

This function generates a picture. It uses a chromosome data and the output results of the segmentation function chrSegment().

Usage

```
chrPlot2(chrData, chrSeg, deb = 100, fin = 1e10)
```

Arguments

chrData	A chromosome data obtained from an output of the function dataSummary(): xinfo2 = dataSummary(af, xinfo, chromosomes, alpha) idx = which(xinfo2$xinfo.s$Chromosome == "chr1") chrData = xinfo2$xinfo.s[idx,]

chrSeg	An output object of the function chrSegment(): chrSeg = chrSegment(chrData, nd = 10)
deb	This parameter allows to specify the start position of the chromosome region for plotting. It can be used for zooming. Negative value will lead to the plot of all chromosome data. deb = 100
fin	This parameter allows to specify the end position of the chromosome region for plot. It can be used for zooming. Negative value will lead to the plot of all chromosome data. thr = 1e7

Value

This function generates a picture on the screen

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
# load CGH data and info files
data(cghData)
rownames(cghData) <- cghData[,1];
data(cghInfo)
noms <- colnames(cghData)
m <- length(noms)
samp  <- noms[2:m]

# associate statistics with probes in the dataset
af <- pfcoMod(cghData, samp, log2.opt = 0, trim.opt = 0.25)

chromosomes <- c(7:9)
alpha <- 0.05

# summarize results for each chromosome
xinfo2 <- dataSummary(af, cghInfo, chromosomes, alpha)

# focused on chromosome 7 data
idx <- which(xinfo2$xinfo.s$Chromosome == "7")
chrData <- xinfo2$xinfo.s[idx, ]

# segment chromosome 7 data
chrSeg <- chrSegment(chrData, nd = 15)

# plot chromosome 7 results
op <- par(mfrow = c(2,1))
chrPlot(chrData, thr = alpha, deb =-1, fin = 3.5e7)
```

```
chrPlot2(chrData, chrSeg, -1, fin = 3.5e7)
par(op)
```

---

chrSegment *Segmentation of a chromosome data*

Description

This function allows to segment a chromosome data

Usage

```
chrSegment(chrData, nd = 10)
```

Arguments

chrData	A chromosome data obtained from an output of the function dataSummary(): xinfo2 = dataSummary(af, xinfo, chromosomes, alpha) idx = which(xinfo2$xinfo.s$Chromosome == "chr1") chrData = xinfo2$xinfo.s[idx,]
nd	The acceptable number of non-detected probes which can separate two significant probes in a segment. Default setting value is 10: nd = 10

Value

This function returns a data frame containing 6 information for each segment

idStart	The start position indexes associated with segments
idEnd	The End position indexes associated with segments
lBounds	The lower bound positions associated with segments
uBounds	The upper bound positions associated with segments
segL2R	The change values associated with segments
segProba	The probabilities associated with segments

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
# load CGH data and info files
data(cghData)
rownames(cghData) <- cghData[,1]
data(cghInfo)
noms <- colnames(cghData)
m <- length(noms)
samp  <- noms[2:m]

# associate statistics with probes in the dataset
af <- pfcoMod(cghData, samp, log2.opt = 0, trim.opt = 0.25)

chromosomes = c(7:9)
alpha <- 0.05

# summarize results for each chromosome
xinfo2 <- dataSummary(af, cghInfo, chromosomes, alpha)

# focused on chromosome 7 data
idx <- which(xinfo2$xinfo.s$Chromosome == "7")
chrData <- xinfo2$xinfo.s[idx, ]

# segment chromosome 7 data
chrSeg <- chrSegment(chrData, nd = 15)

# show first 10 segment results
chrSeg[1:10,]
```

`dataSummary` *Summarization of the detection results for a list of chromosomes*

Description

From an outpout object of the function fcrosMod() or pfcoMod(), the chromosomes information object, the list of chromosomes and a threshold, this function creates two objects containing ordered chromosome data and summary results.

Usage

```
dataSummary(af, xinfo, chromosomes = c(1:22,"X","Y"), alpha = 0.05)
```

Arguments

`af`	An output object of the function fcrosMod() or pfcoMod(): `af` = fcrosMod(xdata, samp, log2.opt, trim.opt) `af` = pfcoMod(xdata, samp, log2.opt, trim.opt)

xinfo	A data frame containing chromosomes information (probe name, gene symbol, chromosome index, start position, end position and the cytoband). These information should appear with the labels ProbeName, GeneSymbol, Chromosome, Start, End, Cytoband. Additional information may be used. Only labels Chromosome, Start and End are mandotory.
chromosomes	A list of chromosomes. Default setting is a list with all chromosomes: chromosomes = (1:22,"X","Y")
alpha	A threshold allowing to select significant probes based on probabilities. Default setting is 0.05 (5% of error) thr = 0.05

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
# load CGH data and info files
data(cghData)
rownames(cghData) <- cghData[,1]
data(cghInfo)
noms <- colnames(cghData)
m <- length(noms)
samp  <- noms[2:m]

# associate statistics with probes in the dataset
af <- pfcoMod(cghData, samp, log2.opt = 0, trim.opt = 0.25)

chromosomes = c(7:9)
alpha <- 0.05

# summarize results for each chromosome
xinfo2 <- dataSummary(af, cghInfo, chromosomes, alpha)

# display the number of significant probes for each chromosome
xinfo2$chrSumm
```

---

fc2Calc *Calculation of fold change using pairwise comparison values*

Description

This is an internal function for using a C code to calculate fold changes using pairwise comparison of samples.

Usage

```
fc2Calc(rvect, n, m, idx, m2)
```

Arguments

rvect	Vector containing the full or reduced matrix with the pairwise comparison of samples results
n	Number of genes or probes in the dataset
m	Number of columns of the full or reduced matrix of pairwise comparison of samples results
idx	Indexes of the columns to keep in the pairwise comparison of samples
m2	Number of columns in the full or reduced matrix of comparison of samples

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
#    data(fdata);
```

---

fcros *Search for differentially expressed genes/probes*

Description

Implementation of a method based on fold change rank ordering statistics for detecting differentially expressed genes in a dataset. This function should be used with two biological conditions dataset (microarray or RNA-seq, ...). Using pairwise combinations of samples from the two biological conditions, fold changes (FC) are calculated. For each combination, the FC obtained are sorted in increasing order and corresponding rank values are associated with genes. Then, a statistic is assigned to the robust average ordered rank values for each gene/probe.

Usage

```
fcros(xdata, cont, test, log2.opt = 0, trim.opt = 0.25)
```

Arguments

xdata	A matrix or a table containing two biological conditions dataset to process for detecting differentially expressed genes. The rownames of xdata are used for the output idnames.
cont	A vector containing the label names of the control samples: cont = c("cont01", "cont02", ...).
test	A vector containing the label names of the test samples: test = c("test01", "test02", "test03", ...).
log2.opt	A scalar equals to 0 or 1. The value 0 (default) means that data in the matrix "xdata" are expressed in a log2 scale: log2.opt = 0
trim.opt	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values of each gene are not used for computing its statistics "ri", i.e. the interquartile range rank values are averaged: trim.opt = 0.25

Details

Label names appearing in the parameters "cont" and "test" should match with some label names in the columns of the data matrix "xdata". It is not necessary to use all label names appearing in the columns of the dataset matrix.

Value

This function returns a data frame containing 9 components

idnames	A vector containing the list of IDs or symbols associated with genes
ri	The average of rank values associated with genes. These values are rank values statistics leading to f-values and p-values.
FC	The fold changes for genes in the dataset. These fold changes are calculated as a ratio of averages from the test and the control samples. Non log scale values are used in the calculation.
FC2	The robust fold changes for genes. These fold changes are calculated as a trimmed mean of the fold changes or ratios obtained from the dataset samples. Non log scale values are used in the calculation.
f.value	The f-values are probabilities associated with genes using the "mean" and the "standard deviation" ("sd") of the statistics "ri". The "mean" and "sd" are used as a normal distribution parameters.
p.value	The p-values associated with genes. These values are obtained from the fold change rank values and one sample t-test.
bounds	Two values, which are the lower and the upper bounds or the minimum and the maximum values of the non standardized "ri".
params	Three values, which are the estimates for the parameters "delta" (average difference between consecutive ordered average of rank values) "mean" (mean value of "ri") and the standard deviation ("sd") of "ri".
params_t	Three values which are theoretical levels for parameters "delta", "mean" and "sd".

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];
cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
trim.opt <- 0.25;

# perform fcros()
af <- fcros(fdata, cont, test, log2.opt, trim.opt);

# now select top 20 down and/or up regulated genes
top20 <- fcrosTopN(af, 20);
alpha1 <- top20$alpha[1];
alpha2 <- top20$alpha[2];
id.down  <- matrix(0, 1);
id.up <- matrix(0, 1);
n <- length(af$FC);
f.value <- af$f.value;

idown <- 1;
iup <- 1;
for (i in 1:n) {
    if (f.value[i] <= alpha1) { id.down[idown] <- i; idown <- idown + 1; }
    if (f.value[i] >= alpha2) { id.up[iup] <- i; iup <- iup + 1; }
}

data.down <- fdata[id.down[1:(idown-1)], ];
ndown <- nrow(data.down);
data.up <- fdata[id.up[1:(iup-1)], ];
nup <- nrow(data.up);


# now plot down regulated genes
t <- 1:20;
op = par(mfrow = c(2,1));
plot(t, data.down[1,2:21], type = "l", col = "blue", xlim = c(1,20),
     ylim = c(0,18), main = "Top down-regulated genes");
```

```
for (i in 2:ndown) {
    lines(t,data.down[i,2:21], type = "l", col = "blue")
}

# now plot down and up regulated genes
plot(t, data.up[1,2:21], type = "l", col = "red", xlim = c(1,20),
    ylim = c(0,18), main = "Top up-regulated genes");
for (i in 2:nup) {
    lines(t, data.up[i,2:21], type = "l", col = "red")
}
par(op)
```

`fcros2` *Search for differentially expressed genes/probes*

Description

Implementation of a method based on fold change or ratio rank ordering statistics for detecting differentially expressed genes. This function should be used with dataset in two separate tables and from two biological conditions datasets (microarray, RNA-seq, ...). Pairwise combinations of samples from the two biological conditions are used to obtain a matrix of fold changes. For each combination, the FCs obtained are sorted in an increasing order and the corresponding rank values are associated with genes/probes. Then, a statistic is associated with each gene/probe.

Usage

```
fcros2(xdata1, xdata2, cont, test, log2.opt = 0, trim.opt = 0.25)
```

Arguments

`xdata1`	A matrix or a table containing two biological conditions dataset to process for detecting differentially expressed genes. The rownames of xdata1 are used for the output idnames.
`xdata2`	A matrix or a table containing two biological conditions dataset to process fordetecting differentially expressed genes: `xdata2`.
`cont`	A vector containing the label names of the control samples: `cont` = c("cont01", "cont02", ...)
`test`	A vector containing the label names of the test samples: `test` = c("test01", "test02", "test03", ...)
`log2.opt`	A scalar equals to 0 or 1. The value 0 (default) means that data in the tables "xdata1" and "xdata2" are expressed in a log2 scale: `log2.opt` = 0
`trim.opt`	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values for a gene are not used for computing the statistics "ri", i.e. the interquartile range rank values are averaged: `trim.opt` = 0.25

Details

The label names appearing in the parameters "cont" and "test" should match some label names in the columns of the data matrices "xdata1" and "xdata2". It is not necessary to use all column label names appearing in matrices "xdata1" and "xdata2". However, it is assumed that the same genes (same IDs or symbol) are used in xdata1 and xdata2.

Value

This function returns a data frame containing 9 components

`idnames`	A vector containing the list of the IDs or symbols associated with genes
`ri`	The average of rank values associated with genes in the datasets. These values are rank statistics leading to f-values and p-values.
`FC`	The fold changes for genes. These fold changes are calculated as a ratio of averages from the test and control samples. Non log scale values are used in the calculation.
`FC2`	The robust fold changes for genes. These fold changes are calculated as a trimmed mean of the fold changes or ratios obtained from the dataset samples using the pairwise comparisons. Non log scale values are used in the calculation.
`f.value`	The f-values are probabilities associated with genes. These values are obtained using the "mean" and the "standard deviation" ("sd") of the statistics "ri". The "mean" and "sd" are used as a normal distribution parameters.
`p.value`	The p-values associated with genes. These values are obtained from the f-values.
`bounds`	Two values which are the lower and the upper bound or the minimum and the maximum values of non standardized "ri".
`params`	Three values, which are the estimates for the parameters "delta" (average difference between consecutive ordered average of rank) "mean" (mean value of variable "ri") and the standard deviation ("sd") of variable "ri".
`params_t`	Three values, which are the theoretical levels for parameters "delta", "mean" and "sd".

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont02", "cont03", "cont04", "cont05",
          "cont06", "cont07", "cont08", "cont09", "cont10");
test <- c("test01", "test02", "test03", "test04", "test05",
          "test06", "test07", "test08", "test09", "test10");
log2.opt <- 0;
trim.opt <- 0.25;

# perform fcros2()
xdata1 <- fdata[,c(2:5, 12:17)];
xdata2 <- fdata[,c(6:11, 18:21)];
rownames(xdata1) <- fdata[,1];
rownames(xdata2) <- fdata[,1];

af2 <- fcros2(xdata1, xdata2, cont, test, log2.opt, trim.opt);

# now select top 20 down and/or up regulated genes
top20 <- fcrosTopN(af2, 20);
alpha1 <- top20$alpha[1];
alpha2 <- top20$alpha[2];
id.down  <- matrix(0,1);
id.up <- matrix(0,1);
n <- length(af2$FC);
f.value <- af2$f.value;

idown <- 1;
iup <- 1;
for (i in 1:n) {
    if (f.value[i] <= alpha1) { id.down[idown] <- i; idown <- idown + 1; }
    if (f.value[i] >= alpha2) { id.up[iup] <- i; iup <- iup + 1; }
}

data.down <- fdata[id.down[1:(idown-1)], ];
ndown <- nrow(data.down);
data.up <- fdata[id.up[1:(iup-1)], ];
nup <- nrow(data.up);

# now plot down regulated genes
t <- 1:20;
op = par(mfrow = c(2,1));
plot(t, data.down[1,2:21], type = "l", col = "blue", xlim = c(1,20),
     ylim = c(0,18), main = "Top down-regulated genes");
for (i in 2:ndown) {
    lines(t,data.down[i,2:21], type = "l", col = "blue")
}

# now plot down and up regulated genes
plot(t, data.up[1,2:21], type = "l", col = "red", xlim = c(1,20),
```

```
    ylim = c(0,18), main = "Top up-regulated genes");
for (i in 2:nup) {
    lines(t, data.up[i,2:21], type = "l", col = "red")
}
par(op)
```

`fcrosFCmat` *Calculation of a matrix of fold changes using pairwise comparisons*

Description

This function is used internally by fcros2() to compute a matrix of fold changes using pairwise comparisons of a two biological conditions dataset.

Usage

```
fcrosFCmat(xdata, cont, test, log2.opt=0, trim.opt=0.25)
```

Arguments

`xdata`	A table containing a two biological conditions dataset to process for obtaining a matrix of fold changes that results from pairwise comparisons of samples. The rownames of xdata are used for the output idnames.
`cont`	A vector containing label names of the control samples: `cont` = c("cont01", "cont02", ...)
`test`	A vector containing label names of the test samples: `test` = c("test01", "test02", "test03", ...)
`log2.opt`	A scalar equals to 0 or 1. The value 0 (default) means that the data values in matrix "xdata" are expressed in a log2 scale: `log2.opt` = 0
`trim.opt`	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values of each gene are not used for computing the statistic "ri" or "u1", i.e. the interquartile rank values are averaged: `trim.opt` = 0.25

Details

The label names appearing in the parameters "cont" and "test" should match some label names of the columns in the table "xdata". The dataset "xdata" can contain other label names not used.

Value

This function returns a data frame containing 3 components

`idnames`	A vector containing the list of IDs or symbols associated with genes
`fcMat`	A matrix of fold changes associated with genes. This matrix is obtained using pairwise comparisons of samples in the dataset "xdata".

FC — The fold changes for genes in the dataset "xdata". These fold changes are calculated as a ratio of averages from the test and the control samples. Non log scale values are used in this calculation.

FC2 — The robust fold changes for genes in the dataset "xdata". These fold changes are calculated as a trimmed mean of the fold changes obtained using pairs of samples. Non log scale values are used in this calculation.

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform fcrosFCmat()
fc <- fcrosFCmat(fdata, cont, test, log2.opt);

# plot histogram of the fold change (log2 scale) in dataset
hist(log2(fc$FC),nclass = 50);
```

---

fcrosMod — *Search for differentially expressed genes or to detect recurrent copy number aberration probes*

Description

Implementation of a method based on fold change rank ordering statistics to search for differentially expressed genes or to detect chromosomal recurrent copy number aberration probes. This function should be used with a matrix of fold changes or ratios from biological dataset (microarray, RNA-seq, ...). The function fcrosMod() is an extention of the function fcros() to a dataset which does not contain replicate samples or to a dataset with one biological condition dataset. Statistics are associated with genes/probes to characterize their change levels.

Usage

```
fcrosMod(fcMat, samp, log2.opt = 0, trim.opt = 0.25)
```

Arguments

fcMat	A matrix containing fold changes or ratios from a biological dataset to process for searching differentially expressed genes or for detecting recurrent copy number aberrations regions. The rownames of fcMat are used for the output idnames.
samp	A vector of sample label names which should appear in the columns of the matrix fcMat: samp.
log2.opt	A scalar equals to 0 or 1. The value 0 (default) means that values in the matrix "fcMat" are expressed in a log2 scale: log2.opt = 0
trim.opt	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values for each gene are not used for computing the statistic "ri", i.e. the interquartile range rank values are averaged: trim.opt = 0.25

Details

The label names appearing in the parameter "samp" should match some label names of the columns in the data matrix "xdata". It is not necessary to use all label names appearing in the columns of the dataset matrix.

Value

This function returns a data frame containing 8 components

idnames	A vector containing the list of IDs or symbols associated with genes
ri	The average of ordered rank values associated with genes in the dataset. These values are rank statistics leading to the f-values and the p-values.
FC2	The robust fold changes for genes in matrix "fcMat". These fold changes are calculated as a trimed mean of the values in "fcMat". Non log scale values are used in this calculation.
f.value	The f-values are probabilities associated with genes using the "mean" and the "standard deviation" ("sd") of values in "ri". The "mean" and "sd" are used as a normal distribution parameters.
p.value	The p-values associated with genes. The p-values are obtained using a one sample t-test on the fold change rank values.
bounds	Two values, which are the lower and the upper bounds or the minimum and the maximum values of the non standardized "ri".
params	Three values, which are the estimates for the parameters "delta" (average difference between consecutive ordered average of rank) "mean" (mean value of the "ri") and the standard deviation ("sd") of the "ri".
params_t	Three values, which are theoretical levels for the parameters "delta", "mean" and "sd".

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
data(fdata);
rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
trim.opt <- 0.25;

# perform fcrosMod()
fc <- fcrosFCmat(fdata, cont, test, log2.opt, trim.opt);
m <- ncol(fc$fcMat)
samp <- paste("Col",as.character(1:m), sep = "");
fc.val <- cbind(data.frame(fc$fcMat))
colnames(fc.val) <- samp
rownames(fc.val) <- fdata[,1]

af <- fcrosMod(fc.val, samp, log2.opt, trim.opt);

# now select top 20 down and/or up regulated genes
top20 <- fcrosTopN(af, 20);
alpha1 <- top20$alpha[1];
alpha2 <- top20$alpha[2];
id.down  <- matrix(0,1);
id.up <- matrix(0,1);
n <- length(af$FC);
f.value <- af$f.value;

idown <- 1;
iup <- 1;
for (i in 1:n) {
    if (f.value[i] <= alpha1) { id.down[idown] <- i; idown <- idown + 1; }
    if (f.value[i] >= alpha2) { id.up[iup] <- i; iup <- iup + 1; }
}

data.down <- fdata[id.down[1:(idown-1)], ];
ndown <- nrow(data.down);
data.up <- fdata[id.up[1:(iup-1)], ];
nup <- nrow(data.up);


# now plot down regulated genes
t <- 1:20;
```

```
op = par(mfrow = c(2,1));
plot(t, data.down[1, 2:21], type = "l", col = "blue", xlim = c(1,20),
     ylim = c(0,18), main = "Top down-regulated genes");
for (i in 2:ndown) {
    lines(t,data.down[i, 2:21], type = "l", col = "blue")
}

# now plot down and up regulated genes
plot(t, data.up[1,2:21], type = "l", col = "red", xlim = c(1,20),
    ylim = c(0,18), main = "Top up-regulated genes");
for (i in 2:nup) {
    lines(t, data.up[i,2:21], type = "l", col = "red")
}
par(op)
```

`fcrosRead` *Read a tab delimited text file*

Description

This function can be used to read a tab delimited text file to be used as input for the functions fcros(), fcros2(), fcrosMod() pfco() or pfcoMod(). You may also use other functions (read.csv(), read.csv2(), read.delim() or read.delim2()) to read your dataset.

Usage

```
fcrosRead(filename)
```

Arguments

`filename`	"filename" is a tab delimited text file to read. The first line of this file should contain labels.

Value

Output is a matrix or a table containing data.

Author(s)

Doulaye Dembele, doulaye@igbmc.fr

Examples

```
# generate random dataset of size 100 x 5
xdata <- matrix(c(rep(0,600)), ncol = 6);

xdata[,2:6] <- matrix(rnorm(500,0,1), ncol = 5);

x0 <- matrix(NA, 100, 0);
```

```
for (i in 1:100) {
    x0[i] <- paste("Obs_",i, sep = "");
}

# set column names
colnames(xdata) <- c("Index", "Col_1", "Col_2", "Col_3", "Col_4", "Col_5");

# save data in a text file
write.table(xdata, file = "test2delete.txt", quote = FALSE, sep = "\t",
       eol = "\n", col.names = TRUE, row.names = FALSE);

# now used fcros.read() to open the file
mydata <- fcrosRead(file = "test2delete.txt");

summary(mydata)
```

`fcrosTopN` *Search for the top N changed genes or probes*

Description

This function allows to seach for the top N differentially expressed genes or changed probes. It uses the output results obtained using one of the following functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod().

Usage

```
fcrosTopN(af, topN)
```

Arguments

`af`	This is an output object obtained using the functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod().
`topN`	The expected number of the top DE genes in the dataset used.

Value

This function returns a data frame containing 2 components

`alpha`	Two threshold values for the down- and the up-regulated allowing to have the top N DE genes
`index`	The indexes of the top N DE genes

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform fcros()
af <- fcros(fdata, cont, test, log2.opt);

# now select top 10 down and/or up regulated genes
top10 <- fcrosTopN(af, 10);

# display thresholds
top10$alpha

# display index of top10 genes
fdata[top10$index, 1]

# display fvalue of the top10 genes
(af$f.value)[top10$index]
```

---

fcrosTtest *Student t-test for detecting differentially expressed genes*

Description

The function uses the basic R t.test() function to perform the Student t-test. It should be used for two biological conditions dataset (microarray, or RNA-seq). The Fold changes, statistics and p-values are returned for each gene in the dataset.

Usage

```
fcrosTtest(xdata, cont, test, log2.opt = 0)
```

Arguments

xdata	A table containing a two biological conditions dataset to process for detecting differentially expressed genes. The rownames of xdata are used for the output idnames.
cont	A vector containing the label names of the control samples: cont = c("cont01", "cont02", ...)

test	A vector containing the label names of the test samples: test = c("test01", "test02", "test03", ...)
log2.opt	A scalar equals to 0 or 1. The value 0 (default) means that data in the matrix "xdata" are expressed in a log2 scale: log2.opt = 0

Details

Label names appearing in the parameters "cont" and "test" should match column label names of the data matrix "xdata". It is not necessary to use all column label names of the dataset "xdata".

Value

idnames	A vector containing the list of IDs or symbols associated with genes
FC	The fold changes for the genes in the dataset.
stat	The Student t-test statistics associated with genes.
p.value	The Student t-test p-values associated with genes.

Author(s)

Doulaye Dembele

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform fcrosTtest()
at <- fcrosTtest(fdata, cont, test, log2.opt);

# now select some differentially expressed genes
id.de  <- matrix(0, 1);
n <- length(at$FC);
for (i in 1:n) {
    if ((at$p.value)[i] <= 0.0005) { id.de <- rbind(id.de, i); }
}

data.de <- fdata[id.de, ];
nde <- nrow(data.de);

# now plot the DE genes
t <- 1:20;
plot(t, data.de[1, 2:21], type = "l", col = "blue", xlim = c(1,20),
     ylim = c(0,18), main = "Down- and up-regulated genes");
for (i in 2:nde) {
    lines(t, data.de[i,2:21], type = "l", col = "blue")
}
```

`fcrosWrite` *Writing the fcros() or pfco() results in a tab delimited text file*

Description

This function creates a tab-delimited text file with the results of the function fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod(). The results are the values associated with the parameters "idnames", "ri", "FC", "FC2", "f.value" and "p.value".

Usage

```
fcrosWrite(af, file = "fcrosResults.txt", thr = 1)
```

Arguments

`af`	An output object of the functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod(): `af = fcros(xdata, cont, test, log2.opt, trim.opt)` `af = pfco(xdata, cont, test, log2.opt, trim.opt)`
`file`	The output file name: `file = "fcrosResults.txt"`
`thr`	A threshold allowing to filter data based on p-values. Default setting is 1 (no filtering) `thr = 1`

Value

This function creates and saves a tab-delimited text file on the disk.

Author(s)

Doulaye Dembele doulaye@igbmc.fr

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1];
cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
trim.opt <- 0.25;
af <- fcros(fdata, cont, test, log2.opt, trim.opt);
af2 <- pfco(fdata, cont, test, log2.opt, trim.opt);

fcrosWrite(af, file = "test2delete_values.txt");
fcrosWrite(af2, file = "test2delete2_values.txt");
```

fdata *Example of dataset for the fcros package.*

Description

This is a synthetic dataset generated using the "madsim" R package (available from the CRAN web site). Two biological conditions data matrix using 10,000 genes and 20 samples (10 control and 10 test samples) were simulated. 108 and 102 genes of this dataset are down- and up-regulated, respectively.

Usage

```
data(fdata)
```

Format

A data frame with 10,000 observations on the following 22 variables.

`index`: a text for the genes, unique ID

`cont01`: a numeric vector with log2 intensities for control sample 01

`cont02`: a numeric vector with log2 intensities for control sample 02

`cont03`: a numeric vector with log2 intensities for control sample 03

`cont04`: a numeric vector with log2 intensities for control sample 04

`cont05`: a numeric vector with log2 intensities for control sample 05

`cont06`: a numeric vector with log2 intensities for control sample 06

`cont07`: a numeric vector with log2 intensities for control sample 07

`cont08`: a numeric vector with log2 intensities for control sample 08

`cont09`: a numeric vector with log2 intensities for control sample 09

`cont10`: a numeric vector with log2 intensities for control sample 10

`test01`: a numeric vector with log2 intensities for test sample 01

`test02`: a numeric vector with log2 intensities for test sample 02

`test03`: a numeric vector with log2 intensities for test sample 03

`test04`: a numeric vector with log2 intensities for test sample 04

`test05`: a numeric vector with log2 intensities for test sample 05

`test06`: a numeric vector with log2 intensities for test sample 06

`test07`: a numeric vector with log2 intensities for test sample 07

`test08`: a numeric vector with log2 intensities for test sample 08

`test09`: a numeric vector with log2 intensities for test sample 09

`test10`: a numeric vector with log2 intensities for test sample 10

`DE_status`: a numeric vector with values -1, 0 and 1. Value 0 is used for no change genes, while -1 and 1 are used for down- and up-regulated genes, respectively.

Details

This dataset is obtained using the microarray data simulation model implemented in the package "madsim". A real microarray data, "madsim_test", was used as seed and the number of the control and the test samples were set to 10. The parameter "sdn" was set to 0.3 and all the other parameters in the madsim package were set to their default settings.

References

Dembele D, A flexible microarray data simulation model. Microarrays, 2013, v.2, n.2, pp.115-130

Examples

```
data(fdata)

rownames(fdata) <- fdata[,1];

op <- par(mfrow = c(2,1));

#  perform MA plot using samples "cont01" and "cont05"
A1 <- 0.5*(fdata$cont01 + fdata$cont05);
M1 <- fdata$cont05 - fdata$cont01;

plot(A1, M1, col="red", xlim=c(2,18), ylim=c(-5,5),
         main="MA plot with two control samples");
lines(x = c(2,18), y = c(0,0), col = "blue")
lines(x = c(2,18), y = c(-1,-1), col = "blue")
lines(x = c(2,18), y = c(1,1), col = "blue")

#  perform MA plot using samples "cont01" and "test05"
A2 <- 0.5*(fdata$cont01 + fdata$test05);
M2 <- fdata$test05 - fdata$cont01;

plot(A2, M2, col="red", xlim=c(2,18), ylim=c(-5,5),
         main="MA plot with one control and one test samples");
lines(x = c(2,18), y = c(0,0), col = "blue")
lines(x = c(2,18), y = c(-1,-1), col = "blue")
lines(x = c(2,18), y = c(1,1), col = "blue")
par(op)
```

---

fvalTopN *Search for the top N changed genes or probes using f-values*

Description

This function allows to seach for the top N differentially expressed genes or changed probes. It uses the f-values obtained using one of the following functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod().

Usage

```
fvalTopN(fval, topN)
```

Arguments

fval	This is a f-values vector obtained using the functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod(): `fval = af$f.value`
topN	The expected number of the top DE genes/probes in the dataset used.

Value

This function returns a data frame containing 2 components

alpha	Two threshold values for the down- and the up-regulated allowing to have the top N DE genes
index	The indexes of the top N DE genes / probes

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);
rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform pfco()
af <- pfco(fdata, cont, test, log2.opt);

# now select top 10 down and/or up regulated genes
top10 <- fvalTopN(af$f.value, 10);

# display thresholds
top10$alpha

# display index of top10 genes
fdata[top10$index, 1]
```

```
# display fvalue of the top10 genes
(af$f.value)[top10$index]
```

`fvalVolcanoPlot` *Performs a volcano plot of the FCROS/PFCO statistics*

Description

This function allows to have a volcano like plot using the output results of the function fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod(): p-values versus robust fold changes (FC2). The p-value are transformed using -log10(), while FC2 are transformed using log2().

Usage

```
fvalVolcanoPlot(af, thr = 0.05)
```

Arguments

`af`	This is an object obtained using the functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod(): `af = fcros(xdata, cont, test)`
`thr`	The threshold to obtain the DE genes in the dataset (red plots): `thr = 0.05`

Value

This function displays on the screen a volcano like plot using the f-values and the robust fold changes (FC2).

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);
rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
```

```
# perform fcros()
af <- fcros(fdata, cont, test, log2.opt);

# Volcano plot
fvalVolcanoPlot(af, thr = 0.01);
```

`histoPlot` *Histogram plot function of the fcros package results*

Description

This function allows to have a histogram plot. It uses the statistics "ri" or "u1" obtained using one of the following functions: fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod().

Usage

```
histoPlot(af, nbins = 50)
```

Arguments

`af`	This is an object obtained using the function fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod()
`nbins`	This parameter is used for the number of bins in the histogram. Default setting is 50: `nbins = 50`

Value

This function plots a histogram on the screen.

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);
rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform fcros() and pfco()
af <- fcros(fdata, cont, test, log2.opt);
af2 <- pfco(fdata, cont, test, log2.opt);

# Histogram plots
op <- par(mfrow = c(1,2))
   histoPlot(af);
   histoPlot(af2);
par(op);
```

`moyStdCalc` *Calculation of the mean and the standard deviation of the full or reduced matrix of sorted ranks*

Description

This is an internal function for using a C code in the calculation of the mean and the standard deviation of the full or reduced matrix with sorted rank values. The calculations are performed for each row.

Usage

```
moyStdCalc(rvect, n, m)
```

Arguments

`rvect`	Vector containing the full or reduced matrix with sorted rank values
`n`	Number of genes or probes in the dataset
`m`	Number of columns of the full or reduced matrix of sorted rank values

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
#    data(fdata);
```

---

pfco *Searching for differentially expressed genes/probes using an approach based on the Perron-Frobenius theorem*

Description

Implementation of a method based on fold change and the Perron theorem for detecting differentially expressed genes in a dataset. This function should be used with two biological conditions dataset (microarray or RNA-seq, ...). Using pairwise combinations of samples from the two biological conditions, fold changes (FC) are calculated. For each combination, the FC obtained are sorted in increasing order and corresponding rank values are associated with genes. Then, a statistic is assigned to the robust average ordered rank values for each gene/probe.

Usage

```
pfco(xdata, cont, test, log2.opt = 0, trim.opt = 0.25)
```

Arguments

`xdata`	A matrix or a table containing two biological conditions dataset to process for detecting differentially expressed genes. The rownames of xdata are used for the output idnames.
`cont`	A vector containing the label names of the control samples: `cont` = c("cont01", "cont02", ...).
`test`	A vector containing the label names of the test samples: `test` = c("test01", "test02", "test03", ...).
`log2.opt`	A scalar equals to 0 or 1. The value 0 (default) means that data in the matrix "xdata" are expressed in a log2 scale: `log2.opt` = 0
`trim.opt`	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values of each gene are not used for computing its statistics "ri", i.e. the interquartile range rank values are averaged: `trim.opt` = 0.25

Details

Label names appearing in the parameter "samp" should match with some label names in the columns of the data matrix "xdata". It is not necessary to use all label names appearing in the columns of the dataset matrix.

Value

This function returns a data frame containing 9 components

idnames	A vector containing the list of IDs or symbols associated with genes
ri	The average of rank values associated with genes. These values are rank values statistics leading to f-values and p-values.
FC	The fold changes for genes in the dataset. These fold changes are calculated as a ratio of averages from the test and the control samples. Non log scale values are used in the calculation.
FC2	The robust fold changes for genes. These fold changes are calculated as a trimmed mean of the fold changes or ratios obtained from the dataset samples. Non log scale values are used in the calculation.
f.value	The f-values are probabilities associated with genes using the "mean" and the "standard deviation" ("sd") of the statistics "ri". The "mean" and "sd" are used as a normal distribution parameters.
p.value	The p-values associated with genes. These values are obtained using a one sample Student t-test on the fold change rank values.
comp	Singular values.
comp.w	Singular values weights.
comp.wcum	Cumulative sum of the singular values weights.

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
data(fdata);
rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
trim.opt <- 0.25;

# perform pfco()
af <- pfco(fdata, cont, test, log2.opt, trim.opt);

# now select top 20 down and/or up regulated genes
top20 <- fcrosTopN(af, 20);
alpha1 <- top20$alpha[1];
alpha2 <- top20$alpha[2];
id.down  <- matrix(0, 1);
```

```
id.up <- matrix(0, 1);
n <- length(af$FC);
f.value <- af$f.value;

idown <- 1;
iup <- 1;
for (i in 1:n) {
    if (f.value[i] <= alpha1) { id.down[idown] <- i; idown <- idown + 1; }
    if (f.value[i] >= alpha2) { id.up[iup] <- i; iup <- iup + 1; }
}

data.down <- fdata[id.down[1:(idown-1)], ];
ndown <- nrow(data.down);
data.up <- fdata[id.up[1:(iup-1)], ];
nup <- nrow(data.up);


# now plot down regulated genes
t <- 1:20;
op = par(mfrow = c(2,1));
plot(t, data.down[1,2:21], type = "l", col = "blue", xlim = c(1,20),
     ylim = c(0,18), main = "Top down-regulated genes");
for (i in 2:ndown) {
    lines(t,data.down[i,2:21], type = "l", col = "blue")
}

# now plot down and up regulated genes
plot(t, data.up[1,2:21], type = "l", col = "red", xlim = c(1,20),
    ylim = c(0,18), main = "Top up-regulated genes");
for (i in 2:nup) {
    lines(t, data.up[i,2:21], type = "l", col = "red")
}
par(op)
```

`pfcoMod` *Searching for differentially expressed genes or detecting recurrent copy number aberration probes using an approach based on the Perron-Frobenius theorem*

Description

Implementation of a method based on fold change rank and the Perron theorem to search for differentially expressed genes or to detect chromosomal recurrent copy number aberration probes. This function should be used with a matrix of fold changes or ratios from biological dataset (microarray, RNA-seq, ...). The function pfcoMod() is an extention of the function pfco() to a dataset which does not contain replicate samples or to a dataset with one biological condition dataset. Statistics are associated with genes/probes to characterize their change levels.

Usage

```
pfcoMod(fcMat, samp, log2.opt = 0, trim.opt = 0.25)
```

Arguments

fcMat	A matrix containing fold changes or ratios from a biological dataset to process for searching differentially expressed genes or for detecting recurrent copy number aberrations regions. The rownames of fcMat are used for the output idnames.
samp	A vector of sample label names which should appear in the columns of the matrix fcMat: samp.
log2.opt	A scalar equals to 0 or 1. The value 0 (default) means that values in the matrix "fcMat" are expressed in a log2 scale: log2.opt = 0
trim.opt	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values for each gene are not used for computing the statistic "ri", i.e. the interquartile range rank values are averaged: trim.opt = 0.25

Details

The label names appearing in the parameter "samp" should match some label names of the columns in the data matrix "xdata". It is not necessary to use all label names appearing in the columns of the dataset matrix.

Value

This function returns a data frame containing 8 components

idnames	A vector containing the list of IDs or symbols associated with genes
ri	The average of ordered rank values associated with genes in the dataset. These values are rank statistics leading to the f-values and the p-values.
FC2	The robust fold changes for genes in matrix "fcMat". These fold changes are calculated as a trimed mean of the values in "fcMat". Non log scale values are used in this calculation.
f.value	The f-values are probabilities associated with genes using the "mean" and the "standard deviation" ("sd") of values in "ri". The "mean" and "sd" are used as a normal distribution parameters.
p.value	The p-values associated with genes. The p-values are obtained from the fold change ranks using a one sample t-test.
comp	Singular values.
comp.w	Singular values weights.
comp.wcum	Cumulative sum of the singular values weights.

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
data(fdata);
rownames(fdata) <- fdata[,1];

cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;
trim.opt <- 0.25;

# perform pfcoMod()
fc <- fcrosFCmat(fdata, cont, test, log2.opt, trim.opt);
m <- ncol(fc$fcMat)
samp <- paste("Col",as.character(1:m), sep = "");
fc.val <- cbind(data.frame(fc$fcMat))
colnames(fc.val) <- samp
rownames(fc.val) <- fdata[,1]

af <- pfcoMod(fc.val, samp, log2.opt, trim.opt);

# now select top 20 down and/or up regulated genes
top20 <- fcrosTopN(af, 20);
alpha1 <- top20$alpha[1];
alpha2 <- top20$alpha[2];
id.down  <- matrix(0,1);
id.up <- matrix(0,1);
n <- length(af$FC);
f.value <- af$f.value;

idown <- 1;
iup <- 1;
for (i in 1:n) {
    if (f.value[i] <= alpha1) { id.down[idown] <- i; idown <- idown + 1; }
    if (f.value[i] >= alpha2) { id.up[iup] <- i; iup <- iup + 1; }
}

data.down <- fdata[id.down[1:(idown-1)], ];
ndown <- nrow(data.down);
data.up <- fdata[id.up[1:(iup-1)], ];
nup <- nrow(data.up);


# now plot down regulated genes
t <- 1:20;
op = par(mfrow = c(2,1));
plot(t, data.down[1, 2:21], type = "l", col = "blue", xlim = c(1,20),
     ylim = c(0,18), main = "Top down-regulated genes");
for (i in 2:ndown) {
    lines(t,data.down[i, 2:21], type = "l", col = "blue")
}

# now plot down and up regulated genes
plot(t, data.up[1,2:21], type = "l", col = "red", xlim = c(1,20),
```

```
    ylim = c(0,18), main = "Top up-regulated genes");
for (i in 2:nup) {
    lines(t, data.up[i,2:21], type = "l", col = "red")
}
par(op)
```

---

pvalTopN *Search for the top N changed genes or probes using p-values*

Description

This function allows to seach for the top N differentially expressed genes or changed probes. It uses the p-value. These p-values can be obtained using one of the following functions fcros(), fcros2(), fcrosMod(), pfco(), pfcoMod() or another statistical method.

Usage

```
pvalTopN(pval, topN)
```

Arguments

pval	This is vector with p-values obtained using the functions fcros(), fcros2(), fcrosMod(), pfco(), pfcoMod() or another statistical method: `pval = af$p.value`
topN	The expected number of the top DE genes/probes in the dataset used: `topN`

Value

This function returns a data frame containing 2 components

alpha	Two threshold values for the down- and the up-regulated allowing to have the top N DE genes
index	The indexes of the top N DE genes

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1]
cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform fcros()
af <- fcros(fdata, cont, test, log2.opt);

# now select top 10 down and/or up regulated genes
top10 <- pvalTopN(af$p.value, 12);

# display thresholds
top10$alpha

# display index of top10 genes
fdata[top10$index, 1]

# display fvalue of the top10 genes
(af$f.value)[top10$index]
```

`pvalVolcanoPlot` *Performs a volcano plot of the FCROS/PFCO statistics*

Description

This function allows to have a volcano like plot using the output results of the function fcros(), fcros2(), fcrosMod(), pfco() or pfcomod(): p-values versus robust fold changes (FC2). The p-value are transformed using -log10(), while FC2 are transformed using log2().

Usage

```
pvalVolcanoPlot(af, thr = 0.05)
```

Arguments

`af`	This is an object obtained using the functions fcros(), fcros2(), fcrosMod(), pfco() or pfcoMod(): `af = fcros(xdata, cont, test)`
`thr`	The threshold to obtain the DE genes in the dataset (red plots): `thr = 0.05`

Value

This function displays on the screen a volcano like plot using the p-values and the robust fold changes (FC2).

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kastner P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
data(fdata);

rownames(fdata) <- fdata[,1]
cont <- c("cont01", "cont07", "cont03", "cont04", "cont08");
test <- c("test01", "test02", "test08", "test09", "test05");
log2.opt <- 0;

# perform fcros()
af <- fcros(fdata, cont, test, log2.opt);

# Volcano plot
pvalVolcanoPlot(af, thr = 1e-6);
```

`rankReads` *This function computes a score to assess the significance of sequencing values*

Description

Implementation of two methods based (1) on the coefficient of variation or (2) on the fold change rank ordering statistics for detecting genes with significant sequencing values (gwssv). A score is obtained for each gene and a threshold allows to select the number of gwssv.

Usage

```
rankReads(xdata, cont, test, meth=0, Ttimes=10, err=0.1, trim.opt=0,
                         rseed=60)
```

Arguments

`xdata`	A matrix or a table containing sequencing dataset. The rownames of xdata is used for the output idnames.
`cont`	A vector containing the label names of the control samples: `cont` = c("cont01", "cont02", ...).

test	A vector containing the label names of the test samples: test = c("test01", "test02", "test03", ...).
meth	This parameter allows to specify the approach to use. The value 0 (defaul) means the coefficient of variation is used. When non-zero value is given, the fcros method used: meth = 0
Ttimes	The number of perturbed data to use. The value 10 (default) means that the dataset is used 20 times and small uniform values are added at each time: Ttimes = 10
err	This is the amount of the uniform values to add to count values. The value 0.1 (default) is used: err = 0.1
trim.opt	A scalar between 0 and 0.5. The value 0.25 (default) means that 25% of the lower and the upper rank values of each gene are not used for computing its statistics "ri", i.e. the inter-quartile range rank values are averaged: trim.opt = 0.25
rseed	This value allow to set the computer random generation seed value in order to be able to have the same results for runs performed at different times: rseed = 58

Details

Label names appearing in the parameters "cont" and "test" should match with some label names in the columns of the data matrix "xdata". It is not necessary to use all label names appearing in the columns of the dataset matrix. For a general purpose dataset, one of these parametere can be empty.

Value

This function returns a data frame containing 10 components when meth=1 and 3 components when meth=0

idnames	A vector containing the list of IDs or symbols associated with genes
score	coefficient of variation (meth=0) or Fisher-Snedecor test p-value (meth=1). Smaller (higher) values are associated with genes with significant (non significant) sequencing values.
moy	trimmed means associated with genes (when meth=0).
ri	The average of rank values associated with genes when meth=1. These values are rank values statistics leading to f-values and p-values (when meth=1).
FC	The fold changes for genes in the dataset. These fold changes are calculated as a ratio of averages from the test and the control samples. Non log scale values are used in the calculation (when meth=1).
FC2	The robust fold changes for genes. These fold changes are calculated as a trimmed mean of the fold changes or ratios obtained from the dataset samples. Non log scale values are used in the calculation (when meth=1).
f.value	The f-values are probabilities associated with genes using the "mean" and the "standard deviation" ("sd") of the statistics "ri". The "mean" and "sd" are used as a normal distribution parameters (when meth=1).
p.value	The p-values associated with genes. These values are obtained from the fold change rank values and one sample t-test (when meth=1).

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, manuscript under preparation

Examples

```
data(bott);
cont <- c("SRX033480", "SRX033488", "SRX033481");
test <- c("SRX033493", "SRX033486", "SRX033494");
n <- nrow(bott);

x2 <- tcnReads(bott[,c(cont,test)])
idx.ok <- (apply(x2, 1, sum) != 0)
xdata <- x2[,c(cont,test)]
rownames(xdata) <- bott[,1]
idx.ok <- (apply(x2, 1, sum) != 0)
tt2 <- sum(idx.ok)

raf10.cv <- rankReads(xdata, cont, test, meth=0)
raf10.pv <- rankReads(xdata, cont, test, meth=1)
score.cv <- -log10(sort(raf10.cv$score))
score.pv <- -log10(sort(raf10.pv$score))
tmp <- scoreThr(score.cv, 2500, 3500)
tmp

tmp <- scoreThr(score.pv, 2500, 3500)
tmp

op <- par(mfrow = c(1,2))
plot(score.cv, xlab = "index of genes",
   ylab = "-log10(sorted(score)", main = "rs.cv", type = "l",
   col = "blue", panel.first = grid())
plot(score.pv, xlab = "index of genes",
   ylab = "-log10(sorted(score)", main = "rs.pv", type = "l",
   col = "blue", panel.first = grid())
par(op)
```

---

rmatCalc *Calculation of the FC matrix using pairwise comparisons*

Description

This is an internal function for using a C code to calculate a vector form of the matrix of fold changes using pairwise comparison of data samples.

Usage

```
rmatCalc(fvect, n, m1, m2)
```

Arguments

fvect	Two biological conditions dataset matrix
n	Number of genes or probes in the dataset
m1	Number of samples in the first biological condition
m2	Number of samples in the second biological condition

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D and Kaster P, Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC bioinformatics, 2014, 15:14

Dembele D and Kastner P, Comment on: Fold change rank ordering statistics: a new method for detecting differentially expressed genes, BMC Bioinformatics, 2016, 17:462

Examples

```
#    data(fdata);
```

---

rmatTrim *Calculation of the reduced matrix containing sorted rank values*

Description

This is an internal function for using a C code for calculating the reduced matrix of sorted rank values.

Usage

```
rmatTrim(rvect, n, m, idx, m2)
```

Arguments

rvect	Vector containing the full matrix with sorted rank values
n	Number of genes or probes in the dataset
m	Number of columns of the full matrix of sorted rank values
idx	Indexes of the columns to keep in the reduced matrix
m2	Number of columns of the reduced matrix

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
#    data(fdata);
```

---

scoreThr *Determine numerically a threshold for the ranking score*

Description

Given a log 10 transformed of the sorted ranking score values (dscore), this function determines numerically a threshold (inflection point) for significance of sequencing value level. User should specify the lower and the upper bounds of the slope region containing the inflection point.

Usage

```
scoreThr(dscore, deb, fin)
```

Arguments

dscore	transformed score values obtained using the function rankReads()
deb	integer for lower bound of the slope region containing the inflection point. User should perfom a plot to determine this value
fin	integer for upper bound of the slope region containing the inflection point. User should perfom a plot to determine this value

Value

This function returns two values: “pos” and “thr” which are the index or the number of genes with significant values and the score (-log10(score)) threshold

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, manuscript under preparation

Examples

```
#    data(fdata);
```

tcnReads *Performs a total count normalization of reads*

Description

Given a data table x with count reads, one column for each sample, this function adjust values in such a way they become comparable between samples. User can specify the maximum value for total count reads to be used for each sample.

Usage

```
tcnReads(x, maxVal=0)
```

Arguments

x	Data table with count reads, one column for one sample
maxVal	Target total number reads for each sample. The value 0 (default) means that the median total of samples is used: maxVal = 0

Value

This function returns a data table of the same size as input

x2	Data table of the same size as x

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Dembele D, manuscript under preparation

Examples

```
#    data(fdata);
```

tprobaCalc *Calculation of the Student one sample test probabilities*

Description

This is an internal function for using a C code to perform a Student one sample test for each row of the full or reduced matrix with sorted rank values.

Usage

```
tprobaCalc(moy, std, n, dl, em)
```

Arguments

moy	Vector containing average of rank values for rows
std	Vector containing standard deviation of rank values for rows
n	Number of genes or probes in the dataset
dl	Degree of freedom in the test. This is equal to the number of the columns in the full or reduced matrix of sorted rank values minus one
em	Expected average rank values for each row

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, Analysis of high biological data using their rank values, Stat Methods Med Res, accepted for publication, 2018

Examples

```
#    data(fdata);
```

varBeta *Compute variance of a beta distribution from data*

Description

This is an internal function. Given a vector with values between 0 and 1 and assumed to come from a beta distribution, this function return the variance of the distribution. A trim parameter allow to have a robust value

Usage

```
varBeta(x, trim.opt)
```

Arguments

x	vector with components between 0 and 1
trim.opt	value between 0 and 0.5 used as trim parameter

Value

This function returns one value equalt to the variance of the best beta distribution of entry x

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Dembele D, manuscript under preparation

Examples

```
#    data(fdata);
```

---

voomReads *Transformation of read count values*

Description

This function allows to transform the count values associated with Sequencing reads. The purpose of this transformation is to allow applying normal-based microarray-like statistical methods to RNA-seq read counts. Log2 values are returned.

Usage

```
voomReads(x, Rm=1e+06)
```

Arguments

x	This is a read counts matrix
Rm	A constant used in the transformation

Value

This function returns a data matrix of the same size as input matrix x. The values of this matrix are expressed in log2 scale.

Author(s)

Doulaye Dembele doulaye@igbmc.fr

References

Charity W Law, Yunshun Chen, Wei Shi and Gordon K Smyth, voom: precision weights unlock linear model analysis tools for RNA-seq read counts,Genome Biology, 2014, 15R29

Examples

```
#   data(fdata);
```

Index