Package 'TCC'

May 27, 2013

Type Package

Title TCC: Differential expression analysis for tag count data with robust normalization strategies

Version 1.1.99

Date 2013-05-20

Author Sun Jianqiang, Tomoaki Nishiyama, Kentaro Shimizu, and Koji Kadota

Maintainer Tomoaki Nishiyama <tomoakin@staff.kanazawa-u.ac.jp>

Description This package provides functions for performing differential expression analysis using differentially expressed gene elimination strategy. A simple unified interface is provided which encapsulates functions to calculate normalization factors and estimate differentially expressed genes defined in edgeR, baySeq, and DESeq packages.
The appropriate combination provided by TCC allows a more robust and accurate estimation performed easily than directly using original packages. Functions to produce simulation data under various conditions and to plot the data are also provided.

Depends R (>= 2.15), methods, DESeq, edgeR, baySeq, ROC

Enhances snow

License GPL-2

URL http://www.iu.a.u-tokyo.ac.jp/~kadota/TCC

R topics documented:

---

arab *Arabidopsis RNA-Seq data set*

Description

This dataset was imported from NBPSeq package and the following explanation is verbatim copy of their explanation:

An RNA-Seq dataset from a pilot study of the defense response of *Arabidopsis* to infection by bacteria. We performed RNA-Seq experiments on three independent biological samples from each of the two treatment groups. The matrix contains the frequencies of RNA-Seq reads mapped to genes in a reference database. Rows correspond to genes and columns correspond to independent biological samples.

Usage

```
data(arab)
```

Format

A 26222 by 6 matrix of RNA-Seq read frequencies.

Details

This dataset was imported from NBPSeq package and the following explanation is verbatim copy of their explanation:

We challenged leaves of *Arabidopsis* with the defense-eliciting Δ*hrcC* mutant of *Pseudomonas syringae* pathovar *tomato* DC3000. We also infiltrated leaves of *Arabidopsis* with 10mM MgCl2 as a mock inoculation. RNA was isolated 7 hours after inoculation, enriched for mRNA and prepared for RNA-Seq. We sequenced one replicate per channel on the Illumina Genome Analyzer (http://www.illumina.com). The length of the RNA-Seq reads can vary in length depending on user preference and the sequencing instrument. The dataset used here are derived from a 36-cycle sequencing reaction, that we trimmed to 25mers. We used an in-house computational pipeline to process, align, and assign RNA-Seq reads to genes according to a reference database we developed for *Arabidopsis*.

References

Di Y, Schafer DW, Cumbie JS, and Chang JH (2011): "The NBP Negative Binomial Model for Assessing Differential Gene Expression from RNA-Seq", Statistical Applications in Genetics and Molecular Biology, 10 (1).

---

calcAUCValue *Calculate AUC value from a TCC-class object*

Description

This function calculates AUC (Area under the ROC curve) value from a TCC-class object for simulation study.

Usage

```
calcAUCValue(tcc)
```

Arguments

tcc	TCC-class object having values in both stat$rank and simulation$trueDEG fields.

Details

This function is generally used after the estimateDE function that estimates p-values (and the derivatives such as the q-values and the ranks) for individual genes based on the statistical model for differential expression (DE) analysis. In case of the simulation analysis, we know which genes are DEGs or non-DEGs in advance and the information is stored in the simulation$trueDEG field of the TCC-class object tcc (i.e., tcc$simulation$trueDEG). The calcAUCValue function calculates the AUC value between the ranked gene list obtained by the estimateDE function and the truth obtained by the simulateReadCounts function. A well-ranked gene list should have a high AUC value (i.e., high sensitivity and specificity).

Value

numeric scalar.

Examples

```
# Analyzing a simulation data for comparing two groups
# (G1 vs. G2) with biological replicates.
# the first 200 genes are DEGs, where 180 are up in G1.
# The DE analysis is performed by an exact test in edgeR coupled
# with the DEGES/edgeR normalization factors.
tcc <- simulateReadCounts(Ngene = 1000, PDEG = 0.2,
                          DEG.assign = c(0.9, 0.1),
                          DEG.foldchange = c(4, 4),
                          replicates = c(3, 3))
```

```
tcc <- calcNormFactors(tcc)
tcc <- estimateDE(tcc)
calcAUCValue(tcc)


# Analyzing a simulation data for comparing two groups
# (G1 vs. G2) without replicates.
# the levels of DE are 3-fold in G1 and 7-fold in G2
# The DE analysis is performed by an negative binomial test in
# DESeq coupled with the DEGES/DESeq normalization factors.
tcc <- simulateReadCounts(Ngene = 1000, PDEG = 0.2,
                          DEG.assign = c(0.9, 0.1),
                          DEG.foldchange = c(3, 7),
                          replicates = c(1, 1))
tcc <- calcNormFactors(tcc)
tcc <- estimateDE(tcc)
calcAUCValue(tcc)
```

---

calcNormFactors *Calculate normalization factors*

Description

This function calculates normalization factors using a specified multi-step normalization method from a TCC-class object. The procedure can generally be described as the $STEP1 - (STEP2 - STEP3)_n$ pipeline.

Usage

```
## S4 method for signature 'TCC'
calcNormFactors(tcc, norm.method = NULL,
                test.method = NULL, iteration = TRUE,
                FDR = NULL, floorPDEG = 0.05,
                dispersion = NULL,
                design = NULL, contrast = NULL, coef = NULL,
                fit0 = NULL, fit1 = NULL, comparison = NULL,
                samplesize = 10000, cl = NULL)
```

Arguments

tcc	TCC-class object.
norm.method	character specifying normalization method that is used in both the STEP1 and STEP3. Possible values are "tmm" for the TMM normalization method implemented in the edgeR package, "edger" (same as "tmm"), and "deseq" for the method implemented in the DESeq package. The default is "tmm" when analyzing the count data with multiple replicates (i.e., $\min(\text{table}(\text{tcc\$group}[, 1])) > 1$) and "deseq" when analyzing the count data without replicates (i.e., $\min(\text{table}(\text{tcc\$group}[, 1])) == 1$)).

test.method	character specifying method for identifying differentially expressed genes (DEGs) used in STEP2. Possible values are "edger", "deseq", and "bayseq" for the DEG identification methods implemented in the edgeR, DESeq, and baySeq, respectively. The default is "edger" when analyzing the count data with multiple replicates (i.e., $\min(\text{table}(\text{tcc\$group}[, 1])) > 1$) and "deseq" when analyzing the count data without replicates (i.e., $\min(\text{table}(\text{tcc\$group}[, 1])) == 1$)).
iteration	logical or numeric value specifying the number of iteration (n) in the proposed normalization pipeline: the $STEP1 - (STEP2 - STEP3)_n$ pipeline. If FALSE or 0 is specified, the normalization pipeline is performed only by the method in STEP1. If TRUE or 1 is specified, the three-step normalization pipeline is performed. Integers higher than 1 indicate the number of iteration in the pipeline.
FDR	numeric value (between 0 and 1) specifying the threshold for determining DEGs after STEP2.
floorPDEG	numeric value (between 0 and 1) specifying the minimum value to be eliminated as potential DEGs before performing STEP3.
dispersion	numeric vector giving the dispersion of all genes for analyzing the count data without replicates when the test.method = "edger" is specified. See the exactTest function in edgeR for details.
design	numeric matrix giving the design matrix for the linear model. See the glmFit function in edgeR for details.
contrast	numeric vector specifying a contrast of the linear model coefficients to be tested equal to zero. See the glmLRT function in edgeR for details.
coef	integer or character vector indicating which coefficients of the linear model are to be tested equal to zero. See the glmLRT function in edgeR for details.
fit0	a formula for creating reduced model described in DESeq. The left hand side must be 'count', the right hand side can involve any column of tcc$group is used as the model frame. See the fitNbinomGLMs function for details.
fit1	a formula for creating full model described in DESeq. The left hand side must be 'count', the right hand side can involve any column of tcc$group is used as the model frame. See the fitNbinomGLMs function for details.
comparison	numeric or character string identifying the columns in the tcc$group[, 1] for analysis. See the group argument in topCounts for details.
samplesize	numeric value specifying the sample size for estimating the prior parameters if test.method = "bayseq". See the getPriors.NB function for details.
cl	'snow' object for using multi processors if test.method = "bayseq". See the getPriors.NB function for details.

Details

The calcNormFactors function is the main function in the TCC package. Since this pipeline employs the DEG identification method at STEP2, our multi-step strategy can eliminate the biased effect of potential DEGs before the second normalization at STEP3. To fully utilize the differentially expressed gene elimination strategy (DEGES), we strongly recommend not to use iteration = 0

or iteration = FALSE. This function internally calls functions implemented in the edgeR, DESeq, and baySeq packages according to the specified parameters.

If the norm.method = "tmm" is specified, the calcNormFactors function implemented in edgeR is used for obtaining the TMM normalization factors at both STEP1 and 3. In case of norm.method = "deseq", the estimateSizeFactors function in DESeq is used. Note that the original estimateSizeFactors function returns the size factors (not normalization factors). Our calcNormFactors function internally converts the size factors into normalization factors that are comparable to the TMM normalization factors.

If the test.method = "edger" is specified, a series of functions for differential expression analysis (estimateCommonDisp, estimateTagwiseDisp, exactTest, etc.) in edgeR are internally used. Similarly, the test.method = "deseq" internally use several functions (estimateDispersions, nbinomTest, etc.) in DESeq and the test.method = "bayseq" internally use two functions (getPriors.NB and getLikelihoods.NB) in baySeq.

Value

TCC-class object containing the normalization factors in the norm.factors field (numerical vector). In other words, the normalization factors in the norm.factors field are populated. The information about normalization, such as execution time, potential DEGs in STEP2 and DEGES pipeline, are storing in DEGES field (i.e., tcc$DEGES).

Examples

```
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)

# calculating normalization factors using the DEGES/edgeR method
# (the TMM-edgeR-TMM pipeline)
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc)
tcc$norm.factors

# calculating normalization factors using the iterative DEGES/edgeR method
# (iDEGES/edgeR) with n = 3
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc, iteration = 3)
tcc$norm.factors

# calculating normalization factors for simulation data without
# replicates
tcc <- simulateReadCounts(replicates = c(1, 1))
tcc <- calcNormFactors(tcc)
tcc$norm.factors
```

---

do_TbT *Calculate normalization factors for raw tag count data using a multi-step normalization strategy called "TbT"*

Description

This method performs TMM normalization, baySeq, and again TMM normalization to infer a good normalization factor as described in Kadota et al. 2012, Algorithms Mol Biol 7:5. This function will be obsoleted. Please use TCC-class based methods.

Usage

```
do_TbT(data, data.cl, sample_num = 10000)
```

Arguments

data	The data matrix to be analysed. Numerical data only
data.cl	A vector describing the data class for columns in data
sample_num	Sample number for baysian estimation

See Also

edgeR

Examples

```
## Not run:
sample <- NBsample()
out <- do_TbT(sample$data, c(1, 1, 1, 2, 2, 2))

## End(Not run)
```

---

estimateDE *Estimate degrees of differential expression (DE) for individual genes*

Description

This function calculates p-values (or the related statistics) for identifying differentially expressed genes (DEGs) from a TCC-class object. estimateDE internally calls a specified method implemented in other R packages.

Usage

```
estimateDE(tcc, test.method = NULL, FDR = NULL,
        dispersion = NULL, fit0 = NULL, fit1 = NULL, design = NULL,
        contrast = NULL, coef = NULL, comparison = NULL,
        samplesize = 10000, cl = NULL)
```

Arguments

tcc	TCC-class object.
test.method	character string specifying method for identifying DEGs. Possible values are "edger", "deseq", and "bayseq" for the DEG identification methods implemented in the edgeR, DESeq, and baySeq, respectively. The default is "edger" when analyzing the count data with replicates (i.e., $\min(\text{table}(\text{tcc\$group}[,1])) > 1$) and "deseq" when analyzing the count data without replicates (i.e., $\min(\text{table}(\text{tcc\$group}[,1])) == 1$)).
FDR	numeric value (between 0 and 1) specifying the threshold for determining DEGs.
dispersion	numeric vector giving the dispersion of all genes for analyzing the count data without replicates when the test.method = "edger" is specified. See the exactTest function in edgeR for details.
design	numeric matrix giving the design matrix for the linear model. See the glmFit function in edgeR for details.
contrast	numeric vector specifying a contrast of the linear model coefficients to be tested equal to zero. See the glmLRT function in edgeR for details.
coef	integer or character vector indicating which coefficients of the linear model are to be tested equal to zero. See the glmLRT function in edgeR for details.
fit0	a formula for creating reduced model described in DESeq. The left hand side must be 'count', the right hand side can involve any column of tcc$group is used as the model frame. See the fitNbinomGLMs function for details.
fit1	a formula for creating full model described in DESeq. The left hand side must be 'count', the right hand side can involve any column of tcc$group is used as the model frame. See the fitNbinomGLMs function for details.
comparison	numeric or character string identifying the columns in the tcc$group[, 1] for analysis. See the group argument in topCounts for details.
samplesize	numeric value specifying the sample size for estimating the prior parameters if test.method = "bayseq". See the getPriors.NB function for details.
cl	'snow' object for using multi processors if test.method = "bayseq". See the getPriors.NB function for details.

Details

estimateDE function is generally used after the calcNormFactors function calculated normalization factors. estimateDE constructs a statistical model for differential expression (DE) analysis with the calculated normalization factors. This function internally calls individual functions implemented in the edgeR, DESeq, and baySeq packages according to the specified parameters.

If the test.method = "edger" is specified, a series of functions for differential expression analysis (estimateCommonDisp, estimateTagwiseDisp, exactTest, etc.) in edgeR are internally called and p-values (and the derivative such as the q-values and the ranks) are calculated.

The test.method = "deseq" internally use several functions (estimateDispersions, nbinomTest, etc.) in DESeq.

The test.method = "bayseq" internally use two functions (getPriors.NB and getLikelihoods.NB) in baySeq.

Different from the edgeR and DESeq in which the calculated p-values are stored in the stat$p.value field of the TCC-class object tcc, baySeq outputs posterior likelihoods instead of p-values. Therefore, the $(1-likelihood)$ values are stored in the corresponding field in case of test.method = "bayseq".

Value

A TCC-class object containing following fields:

stat$p.value	numeric vector of p-values.
stat$q.value	numeric vector of q-values calculated based on the p-values using the p.adjust function with default parameter settings.
stat$rank	gene rank in order of the p-values.
estimatedDEG	numeric vector consisting of 0 or 1 depending on whether each gene is classified as non-DEG or DEG. The threshold for classifying DEGs or non-DEGs is preliminarily given as the FDR argument.

Examples

```
# Analyzing a simulation data for comparing two groups
# (G1 vs. G2) with biological replicates
# The DE analysis is performed by an exact test in edgeR coupled
# with the DEGES/edgeR normalization factors.
# For retrieving the summaries of DE results, we recommend to use
# the getResult function.
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc)
tcc <- estimateDE(tcc, test.method = "edger", FDR = 0.1)
head(tcc$stat$p.value)
head(tcc$stat$q.value)
head(tcc$estimatedDEG)
result <- getResult(tcc)


# Analyzing a simulation data for comparing two groups
# (G1 vs. G2) without replicates
# The DE analysis is performed by an negative binomial (NB) test
# in DESeq coupled with the DEGES/DESeq normalization factors.
data(hypoData)
group <- c(1, 2)
tcc <- new("TCC", hypoData[, c(1, 4)], group)
tcc <- calcNormFactors(tcc)
tcc <- estimateDE(tcc, test.method = "deseq", FDR = 0.1)
```

exactTestafterTbT *exactTest after TMM-baySeq-TMM procedure*

Description

This function perform exact test with edgeR after TMM-baySeq-TMM procedure via do_TbT. This function will be obsoleted. Use TCC-class based methods instead.

Usage

```
exactTestafterTbT(names, counts, group, sample_num=10000)
```

Arguments

names	A vector containing the name of each element eg gene
counts	The data matrix to be analysed. Numerical data only
group	A vector describing the data class for columns in data
sample_num	Sample number for Bayesian estimation

Value

This function return a list of data, norm_f_TbT, counts, and group. The data is a data frame containing the names and counts as the input, table output by the exactTest(), FDR calculated by p.adjust(), and rank by the FDR. norm_f_TbT, Mval, and Aval are calculated by do_TbT(). The counts and group are the copy of the input.

names	As in the input argument.
counts	As in the input argument.
table	As returned from exactTest().
FDR	False discovery rate calculated by p.adjust().
rank_edgeR	The rank of the above.

See Also

do_TbT exactTest p.adjust

Examples

```
## Not run:
sample <- NBsample()
out <- exactTestafterTbT(paste("gene",1:nrow(sample$data), sep=""),
                         sample$data, c(1,1,1,2,2,2))

## End(Not run)
```

filterLowCountGenes *Filter genes from a TCC-class object*

Description

This function takes a TCC object and returns a new TCC object without genes having low count tags across samples. The threshold is configurable with low.count parameter.

Usage

```
filterLowCountGenes(tcc, low.count = 0)
```

Arguments

tcc	TCC-class object.
low.count	numeric value (>= 0) specifying the threshold for filtering genes. The higher value indicates the more numbers of genes to be filtered out.

Value

TCC-class object consisting of genes whose sum of the counts across samples is equal or higher than the low.count value.

Examples

```
# Filtering genes with zero counts across samples (default) from
# a hypothetical count dataset that originally has 1,000 genes
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)
tcc <- new("TCC", hypoData, group)
dim(tcc$count)
tcc <- filterLowCountGenes(tcc)
dim(tcc$count)

# Filtering genes with 10 counts across samples from hypoData
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)
tcc <- new("TCC", hypoData, group)
dim(tcc$count)
tcc <- filterLowCountGenes(tcc, low.count = 10)
dim(tcc$count)
```

getNormalizedData *Obtain normalized count data*

Description

This function generates normalized count data from both original count data and calculated normalization factors.

Usage

```
getNormalizedData(tcc)
```

Arguments

tcc	TCC-class object.

Details

This function is generally used after the calcNormFactors function that calculates normalization factors. The normalized data is calculated using both the original count data stored in the count field and the normalization factors stored in the norm.factors field in the TCC-class object.

Value

A numeric matrix containing normalized count data.

Examples

```
# Note that the hypoData has non-DEGs at 201-1000th rows
nonDEG <- 201:1000
data(hypoData)
summary(hypoData[nonDEG, ])
group <- c(1, 1, 1, 2, 2, 2)

# Obtaining normalized count data after performing the
# DEGES/edgeR normalization method, i.e., DEGES/edgeR-normalized data
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc)
normalized.count <- getNormalizedData(tcc)
summary(normalized.count[nonDEG, ])

# Obtaining normalized count data after performing the
# TMM normalization method (Robinson and Oshlack, 2010),
# i.e., TMM-normalized data
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc, norm.method = "tmm", iteration = FALSE)
normalized.count <- getNormalizedData(tcc)
summary(normalized.count[nonDEG, ])
```

getResult *Obtain the summaries of results after the differential expression analysis*

Description

This function is generally used after the estimateDE function. It retrieves the summaries of differential expression (DE) results from TCC-class object. The retrieved information includes p-values, q-values, coordinates of M-A plot (i.e., M and A values), and so on.

Usage

```
getResult(tcc, sort = FALSE, floor = 0)
```

Arguments

tcc	TCC-class object
sort	logical. If TRUE, the retrieved results are sorted in order of the stat$rank field in the TCC-class object. If FALSE, the results are retrieved by the original order.
floor	numeric scalar specifying a threshold for adjusting low count data.

Value

A data frame object containing following fields:

gene_id	character vector indicating the id of the count unit, usually gene.
a.value	numeric vector of average expression level on log2 scale (i.e., A-value) for each gene across the compared two groups. It corresponds to the x coordinate in the M-A plot.
m.value	numeric vector of fold-change on $\log_2$ scale (i.e., M-value) for each gene between the two groups compared. It corresponds to the y coordinate in the M-A plot.
p.value	numeric vector of p-values.
q.value	numeric vector of q-values calculated based on the p-values using the p.adjust function with default parameter settings.
rank	numeric vector of gene rank in order of the p-values.
estimatedDEG	numeric vector consisting of 0 or 1 depending on whether each gene is classified as non-DEG or DEG. The threshold for classifying DEGs or non-DEGs is preliminarily given when performing estimateDE.

Examples

```
# Obtaining DE results by an exact test in edgeR coupled with
# the DEGES/edgeR normalization factors
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc)
tcc <- estimateDE(tcc, test.method = "edger", FDR = 0.1)
result <- getResult(tcc, sort = TRUE)
head(result)

# Obtaining DE results by an negative binomial test in DESeq
# coupled with the iterative DEGES/DESeq normalization method
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc, norm.method = "deseq", test.method = "deseq")
tcc <- estimateDE(tcc, test.method = "deseq", FDR = 0.1)
result <- getResult(tcc, sort = TRUE)
head(result)
```

---

hypoData *A simulation dataset for comparing two-group tag count data, focusing on RNA-seq*

Description

A simulation dataset, consisting of 1,000 rows (or genes) and 6 columns (or independent biological samples).

Usage

```
data(hypoData)
```

Format

hypoData is a matrix of dimension 1,000 times 6.

Details

This package typically start the differential expression analysis with a count table matrix such as hypoData where each row indicates the gene (or transcript), each column indicates the sample (or library), and each cell indicates the number of counts to the gene in the sample. The first three columns are produced from biological replicates of, for example, Group 1 and the remaining columns are from Group 2; i.e., G1_rep1, G1_rep2, G1_rep3 vs. G2_rep1, G2_rep2, G2_rep3. This data is generated by the simulateReadCounts function with default parameter settings. The first 200 genes are differentially expressed in the two groups. Of these, the first 180 genes are expressed at a higher level in Group 1 (G1) and the remaining 20 genes are expressed at a higher level in G2. Accordingly, the 201-1000th genes are not differentially expressed (non-DEGs). The levels of differential expression (DE) are four-fold in both groups.

Examples

```
# The 'hypoData' is generated by following commands.
tcc <- simulateReadCounts(Ngene = 1000, PDEG = 0.2,
                          DEG.assign = c(0.9, 0.1),
                          DEG.foldchange = c(4, 4),
                          replicates = c(3, 3))
hypoData <- tcc$count
```

---

hypoData_mg *A simulation dataset for comparing three-group tag count data, focusing on RNA-seq*

Description

A simulation dataset, consisting of 1,000 rows (or genes) and 9 columns (or independent biological samples).

Usage

```
data(hypoData_mg)
```

Format

hypoData_mg is a matrix of dimension 1,000 times 9.

Details

The hypoData_mg, a matrix object, is a simulation dataset which consists of 1,000 rows (genes) and 9 columns (samples). Each cell of matrix indicates the number of counts to the gene in the sample. The first three columns are produced from biological replicates of, for example, Group 1, the next three columns are from Group2 and the remaining columns are from Group 3; i.e., G1_rep1, G1_rep2, G1_rep3 vs. G2_rep1, G2_rep2, G2_rep3 vs. G3_rep1, G3_rep2, G3_rep3. This data is generated by the simulateReadCounts function with the following parameters (see Examples). The first 200 genes are differentially expressed among the three groups. Of these, the first 140 genes are expressed at a higher level only in Group 1 (G1), the next 40 genes are expressed at a higher level only in G2 and the last 20 genes are expressed at a higher level only in G3. Accordingly, the 201-1000th genes are not differentially expressed (non-DEGs). The levels of differential expression (DE) are four-fold in each group.

Examples

```
# The 'hypoData_mg' is generated by following commands.
tcc <- simulateReadCounts(Ngene = 1000, PDEG = 0.2,
                          DEG.assign = c(0.7, 0.2, 0.1),
                          DEG.foldchange = c(4, 4, 4),
                          replicates = c(3, 3, 3))
hypoData_mg <- tcc$count
```

MAplot *plot a MA plot*

Description

This method plots a MA plot based on the exactTestafterTbT. This function will be obsoleted.

Usage

```
MAplot(datalist, FDR_threshold = 0.01)
```

Arguments

datalist	The output from exacTestafterTbT
FDR_threshold	Points below the threshold will be plotted in red.

See Also

edgeR

Examples

```
## Not run:
sample <- NBsample()
out <- exactTestafterTbT(paste("gene",1:nrow(sample$data), sep=""),
                  sample$data, c(1,1,1,2,2,2))
MAplot(out)

## End(Not run)
```

NBsample *Sampling from negative biomial distribution*

Description

This methods allow sampling from Negative Binomial distribution with specified proportion of differentially expressed genes having specified level of differential expression in terms of fold change. The proportion of upregulated are also specified. The distribution of original expression levels are generated by resampling real data of *Arabidopsis* RNA-seq data from arab. This function will be obsoleted. Use simulateReadCounts instead.

Usage

```
NBsample(DEG_foldchange=4, repA=3, repB=3, Ngene=3000, PDEG=0.15, PA=0.2)
```

Arguments

DEG_foldchange	Fold change value of differentially expressed genes
repA	Replicate number for sample A
repB	Replicate number for sample B
Ngene	Number of genes to produce
PDEG	Proportion of differentially expressed genes
PA	Proportion of upregulated genes in sample A among differentially expressed genes (DEGs)

---

plot *Plot a log fold-change versus log average expression (so-called M-A plot)*

Description

This function generates a scatter plot of log fold-change (i.e., $M = \log_2 G2 - \log_2 G1$ on the y-axis between Groups 1 vs. 2) versus log average expression (i.e., $A = (\log_2 G1 + \log_2 G2)/2$ on the x-axis) using normalized count data.

Usage

```
## S3 method for class 'TCC'
plot(x, FDR = NULL, median.lines = FALSE, floor = 0,
    groups = NULL, main = NULL,
    xlab = expression(A == (log[2] * G2 + log[2] * G1) / 2),
    ylab = expression(M == log[2] * G2 - log[2] * G1),
    xlim = NULL, ylim = NULL, cex = 0.3, pch = 19,
    col = NULL, col.tag = NULL, ...)
```

Arguments

x	TCC-class object.
FDR	numeric scalar specifying a false discovery rate (FDR) threshold for determining differentially expressed genes (DEGs)
median.lines	logical. If TRUE, horizontal lines specifying the median M values for non-DEGs (black), DEGs up-regulated in Group 1 (G1; blue), and G2 (red) are drawn.
floor	numeric scalar specifying a threshold for adjusting low count data.
groups	numeric vector consists two elements for specifying what two groups should be drawn when data contains more than three groups.
main	character string indicating the plotting title.
xlab	character string indicating the x-label title.

ylab	character string indicating the y-label title.
xlim	numeric vector (consisting of two elements) specifying the range of the x coordinates.
ylim	numeric vector (consisting of two elements) specifying the range of the y coordinates.
cex	numeric scalar specifying the multiplying factor of the size of the plotting points relative to the default (= 0.3).
pch	numeric scalar specifying a symbol or a single character to be used as the default in plotting points.
col	vector specifying plotting color. The default is col = c(1, 4, 2, 5, 6, 7, ...) of color index.
col.tag	numeric vector spacifying the index of col for coloring the points of the genes.
...	further graphical arguments, see plot.default.

Details

This function generates roughly three different M-A plots depending on the conditions for TCC-class objects. When the function is performed just after the new method, all the genes (points) are treated as non-DEGs (the default is black; see Example 1). The simulateReadCounts function followed by the plot function can classify the genes as *true* non-DEGs (black), *true* DEGs up-regulated in Group 1 (G1; blue), and *true* DEGs up-regulated in G2 (red) (see Example 2). The estimateDE function followed by the plot function generates *estimated* DEGs (magenta) and the remaining *estimated* non-DEGs (black).

Genes with normalized counts of 0 in any one group cannot be plotted on the M-A plot because those M and A values cannot be calculated (as log 0 is undefined). Similar to the plotSmear function in edgeR package, plot function plots those points at the left side of the minimum A (i.e., log average expression) value. The x coordinate of those points is the minimum A value minus 1. The y coordinate is calculated as if the 0 count was the minimum observed non-0 count in each group.

Value

A scatter plot to the current graphic device.

Examples

```
# 1.
# M-A plotting just after constructing the TCC class object from
# hypoData. In this case, the plot is generated from hypoData
# that has been scaled in such a way that the library sizes of
# each sample are the same as the mean library size of the
# original hypoData. Note that all points are in black. This is
# because the information about DEG or non-DEG for each gene is
# not indicated.
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)
tcc <- new("TCC", hypoData, group)
plot(tcc)
```

```
normalized.count <- getNormalizedData(tcc)
colSums(normalized.count)
colSums(hypoData)
mean(colSums(hypoData))


# 2.
# M-A plotting of DEGES/edgeR-normalized simulation data.
# It can be seen that the median M value for non-DEGs approaches
# zero. Note that non-DEGs are in black, DEGs up-regulated in
# G1 and G2 are in blue and red.
tcc <- simulateReadCounts()
tcc <- calcNormFactors(tcc)
plot(tcc, median.lines = TRUE)


# 3.
# M-A plotting of DEGES/edgeR-normalized hypoData after performing
# DE analysis.
data(hypoData)
group <- c(1, 1, 1, 2, 2, 2)
tcc <- new("TCC", hypoData, group)
tcc <- calcNormFactors(tcc)
tcc <- estimateDE(tcc, test.method = "edger", FDR = 0.1)
plot(tcc)

# Changing the FDR threshold
plot(tcc, FDR = 0.7)
```

---

plotFCPseudocolor *Create a pseudo-color image of simulation data*

Description

This function creates a pseudo-color image of simulation data regarding the number of differentially expressed genes (DEGs) and the breakdowns for individual groups from a TCC-class object.

Usage

```
plotFCPseudocolor(tcc, main, xlab, ylab)
```

Arguments

tcc	TCC-class object.
main	character string indicating the plotting title.
xlab	character string indicating the *x*-label title.
ylab	character string indicating the *y*-label title.

Details

This function should be used after the simulateReadCounts function that generates simulation data with arbitrary defined conditions. The largest log fold-change (FC) values are in magenta and no-changes are in white.

Examples

```
# Generating a simulation data for comparing two groups
# (G1 vs. G2) with biological replicates.
# the first 200 genes are DEGs, where 180 are up in G1.
tcc <- simulateReadCounts(Ngene = 1000, PDEG = 0.2,
                          DEG.assign = c(0.9, 0.1),
                          DEG.foldchange = c(4, 4),
                          replicates = c(3, 3))
plotFCPseudocolor(tcc)

# Generating a simulation data for comparing three groups
# (G1 vs. G2 vs. G3) with biological replicates.
# the first 300 genes are DEGs, where the 70%, 20%, and 10% are
# up-regulated in G1, G2, G3, respectively. The levels of DE are
# 3-, 10, and 6-fold in individual groups.
tcc <- simulateReadCounts(Ngene = 1000, PDEG = 0.3,
                          DEG.assign = c(0.7, 0.2, 0.1),
                          DEG.foldchange = c(3, 10, 6),
                          replicates = c(3, 3, 3))
plotFCPseudocolor(tcc)
```

---

simulateReadCounts *Generate simulation data from negative binomial (NB) distribution*

Description

This function generates simulation data with a specified condition. It can generate not only all of the simulation data analyzed in Kadota et al., (2012) but also simulation data with more complex design such as two or more groups and/or without replicates.

Usage

```
simulateReadCounts(Ngene = 10000, PDEG = 0.20, DEG.assign = c(0.9, 0.1),
                   DEG.foldchange = NULL, replicates = c(3, 3))
```

Arguments

Ngene	numeric scalar specifying the number of genes.
PDEG	numeric scalar specifying the proportion of differentially expressed genes (DEGs).
DEG.assign	numeric vector specifying the proportions of DEGs up-regulated in individual groups to be compared. The number of element should be the same as the number of groups compared.

DEG.foldchange — numeric vector. The i-th element specifying the degree of FC for Group i. The default is DEG.foldchange = c(4, 4), indicating that the levels of DE are four-fold in both groups.

replicates — numeric vector indicating the numbers of (biological) replicates for individual groups compared.

Details

The empirical distribution of read counts used in this function is calculated from a RNA-seq dataset obtained from *Arabidopsis* data (three biological replicates for both the treated and non-treated samples), the arab object, in NBPSeq package (Di et al., 2011). The overall design about the simulation conditions introduced can be viewed as a pseudo-color image by the plotFCPseudocolor function.

Value

A TCC-class object containing following fields:

count — numeric matrix of simulated count data.

group — numeric vector indicating the numbers of (biological) replicates for individual groups compared.

norm.factors — numeric vector as a placeholder for normalization factors.

stat — list for storing results after the execution of the calcNormFactors (and estimateDE) function.

estimatedDEG — numeric vector as a placeholder for indicating which genes are up-regulated in particular group compared to the others. The values in this field will be populated after the execution of the estimateDE function.

simulation — list containing four fields: trueDEG, DEG.foldchange, PDEG, and group. The trueDEG field (numeric vector) stores information about DEGs: 0 for non-DEG, 1 for DEG up-regulated in Group 1, 2 for DEG up-regulated in Group 2, and so on. The information for the remaining three fields is the same as those indicated in the corresponding arguments.

Examples

```
# Generating a simulation data for comparing two groups
# (G1 vs. G2) without replicates.
# the levels of DE are 3-fold in G1 and 7-fold in G2
tcc <- simulateReadCounts(Ngene = 10000, PDEG = 0.2,
                  DEG.assign = c(0.9, 0.1),
                  DEG.foldchange = c(3, 7),
                  replicates = c(1, 1))
dim(tcc$count)
head(tcc$count)
str(tcc$simulation)
head(tcc$simulation$trueDEG)


# Generating a simulation data for comparing three groups
# (G1 vs. G2 vs. G3) with biological replicates.
```

```
# the first 3000 genes are DEGs, where the 70%, 20%, and 10% are
# up-regulated in G1, G2, G3, respectively. The levels of DE are
# 3-, 10-, and 6-fold in individual groups.
tcc <- simulateReadCounts(Ngene = 10000, PDEG = 0.3,
                          DEG.assign = c(0.7, 0.2, 0.1),
                          DEG.foldchange = c(3, 10, 6),
                          replicates = c(2, 4, 3))
dim(tcc$count)
head(tcc$count)
str(tcc$simulation)
head(tcc$simulation$trueDEG)
```

TCC *A package for differential expression analysis from two-group tag count data with robust normalization strategies*

Description

This package performs differential expression analysis from tag count data that are produced from high-throughput sequencing (HTS) or next generation sequencing (NGS) technology. A notable feature of this package is to provide robust normalization methods whose strategy is to remove data assigned as potential differentially expressed genes (DEGs) before performing data normalization (Kadota et al., 2012).

Details

TCC is a package for differential expression analysis from tag count data, focusing of RNA-seq. This package implements some functions for calculating normalization factors, identifying DEGs, depicting so-called M-A plot, and generating simulation data.

To utilize this package, the count matrix coupled with label information should be stored to a TCC-class object using the new method. All functions (except for some legacy functions) used in this package require this TCC-class object. Using this object, the calcNormFactors function calculates normalization factors and the estimateDE function estimates the degree of differential expression (DE) for individual genes. The estimated normalization factors obtained by using the calcNormFactors function are used within the statistical model for differential analysis in the estimateDE function. Both two functions internally call functions from other packages (edgeR, DESeq, and baySeq) when needed. TCC also provides a useful function simulateReadCounts for generating simulation data with various conditions. Convenient plotting functions are also included.

References

Robinson MD, McCarthy DJ, Smyth GK: edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 2010, 26(1): 139-140

Hardcastle TJ, Kelly KA: baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinformatics 2010, 11: 422

Anders S, Huber W: Differential expression analysis for sequence count data. Genome Biol. 2010, 11: R106

Kadota K, Nishiyama T, Shimizu K: A normalization strategy for comparing tag count data. Algorithms Mol Biol. 2012, 7:5

See Also

TCC-class

TCC-class *A container for storing information used in TCC*

Description

This is the container class for TCC. This class initially contains count data matrix and some information for the analysis of count data. It also provides further fields that are populated during the analysis.

Details

This class is implemented as a R5 reference class. Thus the method call to this class object change the content of the object. Functions calling such methods copies the object prior to calling the method to keep the semantics of functional programming. This class can be created by the generic new function with the data for (i.e., count and group) fields.

The values (defaults to all 1) in the norm.factors field will be changed after performing the calcNormFactors function. The stat field stores (i) execution time for calculating normalization factors after performing the calcNormFactors function, and (iii) statistics (i.e., p-value, q-value, and rank) related to the degrees of differential expression for individual genes after performing the estimateDE function. The estimatedDEG field stores information about which genes are called significantly highly expressed in one group. The threshold for determining the differentially expressed genes (DEGs) is preliminarily indicated when performing the estimateDE function. The simulation field stores parameters of the simulation. The information in this field is generated by the simulateReadCounts function.

Fields

This class contains the following fields:

count numeric matrix containing count data.

gene_id character vector indicating the id of the count unit, usually gene.

group data frame indicating the experimental group for each sample (or library).

norm.factors numeric vector containing normalization factors.

DEGES list for storing the information about normalization steps.

stat list for storing results after the execution of the calcNormFactors and estimateDE functions.

estimatedDEG numeric vector as a placeholder for indicating which genes are expressed higher in particular group compared to the others. The values in this field will be populated after the execution of the estimateDE function.

simulation list. This field is only used for analyzing simulation data.

Examples

```
tcc <- simulateReadCounts()

# Check the TCC class object.
tcc

# Check the fields of TCC class object.
names(tcc)
head(tcc$count)

# Check the normalization factors.
tcc <- calcNormFactors(tcc)
tcc$norm.factors

# Check the p-values and q-values.
tcc <- estimateDE(tcc)
tcc

# Compare the breakdowns of estimated DEGs with the truth.
head(tcc$estimatedDEG)
head(tcc$simulation$trueDEG)

# M-A plotting.
plot(tcc)
```

Index