Package ‘EBSeq’

May 29, 2024

Type Package
Title An R package for gene and isoform differential expression analysis of RNA-seq data
Version 2.2.0
Date 2015-12-8
Depends blockmodeling, gplots, testthat, R (>= 3.0.0)
Description Differential Expression analysis at both gene and isoform level using RNA-seq data
License Artistic-2.0
LazyLoad yes
Collate 'MedianNorm.R' 'GetNg.R' 'beta.mom.R' 'EBTest.R'
 'GetPatterns.R' 'EBMultiTest.R' 'PostFC.R' 'GetPPMat.R'
 'GetMultiPP.R' 'GetMultiFC.R' 'PlotPostVsRawFC.R' 'crit_fun.R'
 'DenNHist.R' 'GetNormalizedMat.R' 'PlotPattern.R'
 'PolyFitPlot.R' 'QQP.R' 'QuantileNorm.R' 'RankNorm.R'
 'GetDEResults.R' 'EBSeqTest.R' 'likefun.R' 'likefunMulti.R'
 'LogN.R' 'LogNMulti.R' 'f0.R' 'f1.R' 'GetSelectedPatterns.R'
BuildVignettes no
Imports Rcpp (>= 0.12.11), RcppEigen (>= 0.3.2.9.0)
LinkingTo Rcpp, RcppEigen, BH
SystemRequirements c++11
Roxygen list(wrap=FALSE)
RoxygenNote 7.1.0
biocViews ImmunoOncology, StatisticalMethod, DifferentialExpression,
 MultipleComparison, RNASEq, Sequencing
NeedsCompilation no
git_url https://git.bioconductor.org/packages/EBSeq
git_branch RELEASE_3_19
git_last_commit 95b79f2
git_last_commit_date 2024-04-30
Repository  Bioconductor 3.19
Date/Publication  2024-05-29
Author  Xiuyu Ma [cre, aut],
        Ning Leng [aut],
        Christina Kendziorski [ctb],
        Michael A. Newton [ctb]
Maintainer  Xiuyu Ma <watsonforfun@gmail.com>

Contents

EBSeq_NingLeng-package ........................................... 3
beta.mom ......................................................... 4
crit_fun ......................................................... 5
DenNHist ........................................................ 6
EBMultiTest ....................................................... 7
EBSeqTest ........................................................ 9
EBTest ........................................................... 10
f0 ................................................................. 12
f1 ................................................................. 13
GeneMat .......................................................... 14
GetDEResults ...................................................... 15
GetMultiFC ......................................................... 17
GetMultiPP ......................................................... 18
GetNg ............................................................ 19
GetNormalizedMat ............................................... 21
GetPatterns ....................................................... 22
GetPPMat ........................................................ 22
GetSelectedPatterns ............................................ 23
IsoList .......................................................... 24
IsoMultiList ..................................................... 25
Likefun .......................................................... 25
LikefunMulti ..................................................... 26
LogN .............................................................. 27
LogNMult ......................................................... 28
MedianNorm ......................................................... 29
MultiGeneMat ..................................................... 30
PlotPattern ....................................................... 31
PlotPostVsRawFC .................................................. 31
PolyFitPlot ....................................................... 32
PostFC .......................................................... 34
QQP .............................................................. 35
QuantileNorm ..................................................... 36
RankNorm ........................................................ 37

Index  39
EBSeq_NingLeng-package

EBSeq: RNA-Seq Differential Expression Analysis on both gene and isoform level

Description

In 'EBSeq_NingLeng-package,' a Negative Binomial-beta model was built to analyze the RNASeq data. We used the empirical bayes method and EM algorithm.

Details

Package: EBSeq_NingLeng
Type: Package
Version: 1.0
Date: 2011-06-13
License: What license is it under?
LazyLoad: yes

Author(s)

Ning Leng, Xiuyu Ma, Christina Kendziorski, Michael A. Newton
Maintainer: Ning Leng <lengning1@gmail.com> Xiuyu Ma <watsonforfun@gmail.com>

References


See Also

EBTest, EBMultiTest

Examples

data(GeneMat)
GeneMat.small = GeneMat[c(1:10,511:550),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data=GeneMat.small,
Conditions=as.factor(rep(c("C1","C2"), each=5)),
sizeFactors=Sizes, maxround=5)
beta.mom  

Fit the beta distribution by method of moments

Description

'beta.mom' fits the beta distribution by method of moments.

Usage

beta.mom(qs.in)

Arguments

qs.in  
A vector contains the numbers that are assumed to follow a beta distribution.

Value

alpha.hat  
Returns the estimation of alpha.

beta.hat  
Returns the estimation of beta.

Author(s)

Ning Leng

References


See Also

DenNHist, DenNHistTable

Examples

#tmp = rbeta(5, 5, 100)
#param = beta.mom(tmp)
crit_fun

Calculate the soft threshold for a target FDR

Description

'crit_fun' calculates the soft threshold for a target FDR.

Usage

crit_fun(PPEE, thre)

Arguments

PPEE 
  The posterior probabilities of being EE.

thre 
  The target FDR.

Details

Regarding a target FDR alpha, both hard threshold and soft threshold could be used. If the hard
threshold is preferred, user could simply take the transcripts with PP(DE) greater than (1-alpha).
Using the hard threshold, any DE transcript in the list is with FDR <= alpha.

If the soft threshold is preferred, user could take the transcripts with PP(DE) greater than crit_fun(PPEE, alpha). Using the soft threshold, the list of DE transcripts is with average FDR alpha.

Value

The adjusted FDR threshold of target FDR.

Author(s)

Ning Leng

References


Examples

data(GeneMat)
GeneMat.small = GeneMat[c(1:10, 500:600),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small, 
  Conditions = as.factor(rep(c("C1","C2"), each=5)), 
  sizeFactors = Sizes, maxround = 5)
PP = GetPMMat(EBOut)
DEfound = rownames(PP)[which(PP[,"PPDE"] >= 0.95)]
DenNHist

Density plot to compare the empirical q's and the simulated q's from the fitted beta distribution.

Description

'DenNHist' gives the density plot that compares the empirical q's and the simulated q's from the fitted beta distribution.

Usage

DenNHist(EBOut, GeneLevel = F)

Arguments

- **EBOut**: The output of EBTest or EBMultiTest.
- **GeneLevel**: Indicate whether the results are from data at gene level.

Value

For data with n1 conditions and n2 uncertainty groups, n1*n2 plots will be generated. Each plot represents a subset of the data. The empirical estimation of q’s will be represented as blue histograms and the density of the fitted beta distribution will be represented as the green line.

Author(s)

Ning Leng

References


See Also

beta.mom, QQP, EBTest, EBMultiTest
**Examples**

```r
data(GeneMat)
GeneMat.small = GeneMat[c(500:1000),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small, Conditions = as.factor(rep(c("C1","C2"), each=5)),
sizeFactors = Sizes, maxround = 5)
par(mfrow = c(2,2))
DenNHist(EBOut)
```

**Description**

'EBMultiTest' is built based on the assumption of NB-Beta Empirical Bayes model. It utilizes the EM algorithm to give the posterior probability of the interested patterns.

**Usage**

```r
EBMultiTest(Data, NgVector = NULL, Conditions, sizeFactors, uc = 0, AllParti = NULL, fast = T,
Alpha = NULL, Beta = NULL, Qtrm = 1, QtrmCut = 0, maxround = 50,
step1 = 1e-06, step2 = 0.01, thre = log(2), sthre = 0,
filter = 10, stopthre = 1e-04, nequal = 2)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>A data matrix contains expression values for each transcript (gene or isoform level). In which rows should be transcripts and columns should be samples.</td>
</tr>
<tr>
<td>NgVector</td>
<td>A vector indicates the uncertainty group assignment of each isoform. e.g. if we use number of isoforms in the host gene to define the uncertainty groups, suppose the isoform is in a gene with 2 isoforms, Ng of this isoform should be 2. The length of this vector should be the same as the number of rows in Data. If it’s gene level data, Ngvector could be left as NULL.</td>
</tr>
<tr>
<td>Conditions</td>
<td>A vector indicates the condition in which each sample belongs to.</td>
</tr>
<tr>
<td>sizeFactors</td>
<td>The normalization factors. It should be a vector with lane specific numbers (the length of the vector should be the same as the number of samples, with the same order as the columns of Data).</td>
</tr>
<tr>
<td>uc</td>
<td>number of unceratian positions, unit levels</td>
</tr>
<tr>
<td>AllParti</td>
<td>user specified set of partitions, a matrix, with each row represent a partition</td>
</tr>
<tr>
<td>fast</td>
<td>boolean indicator whether to use fast EBSeq or full EBSeq</td>
</tr>
<tr>
<td>Alpha</td>
<td>start value of hyper parameter alpha</td>
</tr>
<tr>
<td>Beta</td>
<td>start value of hyper parameter beta</td>
</tr>
</tbody>
</table>
EBMultiTest

Qtrm, QtrmCut: Transcripts with Qtrm th quantile \( \leq \) QtrmCut will be removed before testing. The default value is Qtrm = 1 and QtrmCut = 0. By default setting, transcripts with all 0’s won’t be tested.

maxround: Number of iterations. The default value is 50. Users should always check the convergency by looking at the Alpha and Beta in output. If the hyper-parameter estimations are not converged in 50 iterations, larger number is suggested.

step1, step2: stepsize for gradient ascent of alpha and beta

thre: threshold for determining the state of a position

sthre: shrinkage threshold for iterative pruning during the EM updates

filter: filter threshold for low expression units

stopthre: stopping threshold for EM

nequal: when there is a chain of equal states with the number of equal states bigger than nequal, equalhandle algorithm will be used to further checking the homogeneity between the group means

Value

Alpha: Fitted parameter alpha of the prior beta distribution.

Beta: Fitted parameter beta of the prior beta distribution.

P: Global proportion of DE patterns.

RList: The fitted values of r for each transcript.

MeanList: The mean of each transcript (across conditions).

VarList: The variance of each transcript (across conditions).

QList: The fitted q values of each transcript within the two conditions

Mean: The mean of each transcript within the two conditions (adjusted by normalization factors).

Var: The estimated variance of each transcript within the two conditions (adjusted by normalization factors).

PoolVar: The variance of each transcript (The pooled value of within condition EstVar).

DataNorm: Normalized expression matrix.

Iso: same as NgVector

AllZeroIndex: The transcript with expression 0 for all samples (which are not tested).

PPMat: The Posterior Probability of following each pattern (columns) for each transcript (rows). Transcripts with expression 0 for all samples are not shown in this matrix.

AllParti: selected patterns

PPMatWith0: The Posterior Probability of following each pattern (columns) for each transcript (rows). Transcripts with expression 0 for all samples are shown in this matrix with PP(any_pattern)=NA. The transcript order is exactly the same as the order of the input data.

Conditions: The input conditions.

NumUC: The number of uncertain positions at each unit
EBSeqTest

Author(s)
Ning Leng, Xiuyu Ma

References

See Also
EBTest, GetMultiPP, GetMultiFC

Examples
```r
data(MultiGeneMat)
Conditions = c("C1","C1","C2","C2","C3","C3")
MultiSize = MedianNorm(MultiGeneMat)
MultiOut = EBMultiTest(MultiGeneMat,Conditions=Conditions,uc = 2,
sizeFactors=MultiSize)
MultiPP = GetMultiPP(MultiOut)
```

EBSeqTest

EBSeq core

Description
core function of EBSeq computation. Users are expected to use the wrappers, 2 conditions scenario, using EBTest, more than 2 conditions, using EBMultiTest

Usage
```r
EBSeqTest(data, conditions, uc, AllParti = NULL, iLabel = 1, sizefactor = 1,
iter = 50, alpha = 0.4, beta = 0, step1 = 1e-06, step2 = 0.01,
thre = log(2), sthre = 0.001, filter = 10, stopthre = 0.001, nequal = 2)
```

Arguments
data
A data matrix contains expression values for each transcript (gene or isoform level). In which rows should be transcripts and columns should be samples. For single cell data, normalized counts are required

conditions
condition label for samples

uc
number of unceratin positions, unit level

AllParti
user specified set of partitions

iLabel
label for isoform, indicating how beta are shared among units
sizefactor  The normalization factors. It should be a vector with lane specific numbers (the length of the vector should be the same as the number of samples, with the same order as the columns of Data).
iter          maximum iteration step of EM
alpha         start value of hyper parameter alpha
beta          start value of hyper parameter beta
step1         stepszie for gradient ascent of alpha
step2         stepszie for gradient ascent of beta
thre          threshold for determining the state of a position
sthre         shrinkage threshold for iterative pruning during the EM updates
filter        filter threshold for low expression units
stopthre      stopping threshold for EM
nequal        when there is a chain of equal states with the number of equal states bigger than nequal, equalhandle algorithm will be used to further checking the homogeneity between the group means

Value

a list containing selected DE patterns and their posterior probabilities, values for alpha and beta, some moments of the data

---

**EBTest**

Using EM algorithm to calculate the posterior probabilities of being DE

---

Description

Base on the assumption of NB-Beta Empirical Bayes model, the EM algorithm is used to get the posterior probability of being DE.

Usage

```r
EBTest(Data, NgVector = NULL, Conditions, sizeFactors, fast = T,
       Alpha = NULL, Beta = NULL, Qtrm = 1, QtrmCut = 0, maxround = 50,
       step1 = 1e-06, step2 = 0.01, thre = log(2), sthre = 0,
       filter = 10, stopthre = 1e-4)
```

Arguments

- **Data** A data matrix contains expression values for each transcript (gene or isoform level). In which rows should be transcripts and columns should be samples.
- **NgVector** A vector indicates the uncertainty group assignment of each isoform. e.g. if we use number of isoforms in the host gene to define the uncertainty groups, suppose the isoform is in a gene with 2 isoforms, Ng of this isoform should be 2. The length of this vector should be the same as the number of rows in Data. If it’s gene level data, Ngvector could be left as NULL.
Conditions  A factor indicates the condition which each sample belongs to.

sizeFactors  The normalization factors. It should be a vector with lane specific numbers (the length of the vector should be the same as the number of samples, with the same order as the columns of Data).

fast  boolean indicator whether to use fast EBSeq or full EBSeq

Alpha  start value of hyper parameter alpha

Beta  start value of hyper parameter beta

Qtrm, QtrmCut  Transcripts with Qtrm th quantile ≤ QtrmCut will be removed before testing. The default value is Qtrm = 1 and QtrmCut=0. By default setting, transcripts with all 0’s won’t be tested.

maxround  Number of iterations. The default value is 50. Users should always check the convergency by looking at the Alpha and Beta in output. If the hyper-parameter estimations are not converged in 50 iterations, larger number is suggested.

step1  stepsize for gradient ascent of alpha

step2  stepsize for gradient ascent of beta

thre  threshold for determining the state of a position

sthre  shrinkage threshold for iterative pruning during the EM updates

filter  filter threshold for low expression units

stopthre  stopping threshold for EM

Details

For each transcript gi within condition, the model assumes: \( X_{gislmu, gi} \sim NB \left( r_{gi0} \times l_s, q_{gi} \right) \)

\( q_{gi} | alpha, beta^N_g \sim Beta \left( alpha, beta^N_g \right) \) In which the \( l_s \) is the sizeFactors of samples.

The function will test "H0: \( q_{gi}^{C1} = q_{gi}^{C2} \)" and "H1: \( q_{gi}^{C1} \neq q_{gi}^{C2} \)."

Value

Alpha  Fitted parameter alpha of the prior beta distribution.

Beta  Fitted parameter beta of the prior beta distribution.

P  Global proportion of DE patterns.

RList  The fitted values of r for each transcript.

MeanList  The mean of each transcript (across conditions).

VarList  The variance of each transcript (across conditions).

QList  The fitted q values of each transcript within the two conditions

Mean  The mean of each transcript within the two conditions (adjusted by normalization factors).

Var  The estimated variance of each transcript within the two conditions (adjusted by normalization factors).

PoolVar  The variance of each transcript (The pooled value of within condition EstVar).

DataNorm  Normalized expression matrix.
AllZeroIndex  The transcript with expression 0 for all samples (which are not tested).
Iso  same as NgVector
PPMat  A matrix contains posterior probabilities of being EE (the first column) or DE (the second column). Rows are transcripts. Transcripts with expression 0 for all samples are not shown in this matrix.
AllParti selected patterns
PPMatWith0  A matrix contains posterior probabilities of being EE (the first column) or DE (the second column). Rows are transcripts. Transcripts with expression 0 for all samples are shown as PP(EE) = PP(DE) = NA in this matrix. The transcript order is exactly the same as the order of the input data.
Conditions  The input conditions.

Author(s)
Ning Leng, Xiuyu Ma

References

See Also
EBMultiTest, PostFC, GetPPMat

Examples
data(GeneMat)
str(GeneMat)
Sizes = MedianNorm(GeneMat)
EBOut = EBTest(Data=GeneMat, Conditions=as.factor(rep(c("C1","C2"),each=5)),
    sizeFactors = Sizes)
PP = GetPPMat(EBOut)

---

f0  

The Prior Predictive Distribution of being EE

Description
'f0' gives the Prior Predictive Distribution of being EE.

Usage
f0(Input, AlphaIn, BetaIn, EmpiricalR, NumOfGroups, log)
Arguments

Input Expression Values.
AlphaIn, BetaIn, EmpiricalR
The parameters estimated from last iteration of EM.
NumOfGroups How many transcripts within each Ng group.
log If true, will give the log of the output.

Value
The function will return the prior predictive distribution values of being EE.

Author(s)
Ning Leng

References

See Also
f1

Examples

```
#f1(matrix(rnorm(100,100,1),ncol=10), .5, .6,
# matrix(rnorm(100,200,1),ncol=10), 100, TRUE)
```

Description
The Prior Predictive Distribution of being DE

Usage
f1(Input1, Input2, AlphaIn, BetaIn, EmpiricalRSP1,
EmpiricalRSP2, NumOfGroup, log)
Arguments

Input1  Expressions from Condition1.
Input2  Expressions from Condition2.
AlphaIn, BetaIn, EmpiricalRSP1, EmpiricalRSP2
       The parameters estimated from last iteration of EM.
NumOfGroup  How many transcripts within each Ng group.
log  If true, will give the log of the output.

Value

The function will return the prior predictive distribution values of being DE.

Author(s)

Ning Leng

References


See Also

f0

Examples

#f1(matrix(rnorm(100,100,1),ncol=10),
# matrix(rnorm(100,100,1),ncol=10), .5, .6,
# matrix(rnorm(100,200,1),ncol=10),
# matrix(rnorm(100,200,1),ncol=10), 100, TRUE)

GenMat

The simulated data for two condition gene DE analysis

Description

'GenMat' gives the simulated data for two condition gene DE analysis.

Usage

data(GenMat)
GetDEResults

Source

See Also
IsoList

Examples

```r
data(GeneMat)
```

<table>
<thead>
<tr>
<th>GetDEResults</th>
<th>Obtain Differential Expression Analysis Results in a Two-condition Test</th>
</tr>
</thead>
</table>

## Description

Obtain DE analysis results in a two-condition test using the output of EBTest()

## Usage

```r
GetDEResults(EBPrelim, FDR=0.05, Method="robust", FDRMethod="hard", Threshold.FC=0.7, Threshold.FCRatio=0.3, SmallNum=0.01)
```

## Arguments

- **EBPrelim**: Output from the function EBTest().
- **FDR**: Target FDR, default is 0.05.
- **FDRMethod**: "hard" or "soft". Giving a target FDR alpha, either hard threshold and soft threshold may be used. If the hard threshold is preferred, DE transcripts are defined as the the transcripts with PP(DE) greater than (1-alpha). Using the hard threshold, any DE transcript in the list has FDR <= alpha. If the soft threshold is preferred, the DE transcripts are defined as the transcripts with PP(DE) greater than crit_fun(PPEE, alpha). Using the soft threshold, the list of DE transcripts has average FDR alpha. Based on results from our simulation studies, hard thresholds provide a better-controlled empirical FDR when sample size is relatively small (Less than 10 samples in each condition). User may consider the soft threshold when sample size is large to improve power.
- **Method**: "robust" or "classic". Using the "robust" option, EBSeq is more robust to genes with outliers and genes with extremely small variances. Using the "classic" option, the results will be more comparable to those obtained by using the GetPPMat() function from earlier version (<= 1.7.0) of EBSeq. Default is "robust".
Threshold_FC

Threshold for the fold change (FC) statistics. The default is 0.7. The FC statistics are calculated as follows. First the posterior FC estimates are calculated using PostFC() function. The FC statistics is defined as \( \exp(-\log \text{posterior FC}) \) and therefore is always less than or equal to 1. The default threshold was selected as the optimal threshold learned from our simulation studies. By setting the threshold as 0.7, the expected FC for a DE transcript is less than 0.7 (or greater than 1/0.7=1.4). User may specify their own threshold here. A higher (less conservative) threshold may be used here when sample size is large. Our simulation results indicated that when there are more than or equal to 5 samples in each condition, a less conservative threshold will improve the power when the FDR is still well-controlled. The parameter will be ignored if Method is set as "classic".

Threshold_FCRatio

Threshold for the fold change ratio (FCRatio) statistics. The default is 0.3. The FCRatio statistics are calculated as follows. First we get another revised fold change statistic called Median-FC statistic for each transcript. For each transcript, we calculate the median of normalized expression values within each condition. The MedianFC is defined as \( \exp(-\log((C1\text{Median}+\text{SmallNum})/(C2\text{Median}+\text{SmallNum}))) \). Note a small number is added to avoid Inf and NA. See SmallNum for more details. The FCRatio is calculated as \( \exp(-\log(FC\text{statistics/MedianFC})) \). Therefore it is always less than or equal to 1. The default threshold was selected as the optimal threshold learned from our simulation studies. By setting the threshold as 0.3, the FCRatio for a DE transcript is expected to be larger than 0.3.

SmallNum

When calculating the FCRatio (or Median-FC), a small number is added for each transcript in each condition to avoid Inf and NA. Default is 0.01.

Details

GetDEResults() function takes output from EBTest() function and output a list of DE transcripts under a target FDR. It also provides posterior probability estimates for each transcript.

Value

- DEfound: A list of DE transcripts.
- PPMat: Posterior probability matrix. Transcripts are following the same order as in the input matrix. Transcripts that were filtered by magnitude (in EBTest function), FC, or FCR are assigned with NA for both PPDE and PPEE.
- Status: Each transcript will be assigned with one of the following values: "DE", "EE", "Filtered: Low Expression", "Filtered: Fold Change" and "Filtered: Fold Change Ratio". Transcripts are following the same order as in the input matrix.

Author(s)

Ning Leng, Yuan Li

References

Bayes hierarchical model for inference in RNA-seq experiments. Bioinformatics (2013)

See Also
EBTest

Examples

data(GeneMat)
str(GeneMat)
GeneMat.small = GeneMat[c(1:10,511:550),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small, Conditions = as.factor(rep(c("C1","C2"), each = 5)), sizeFactors = Sizes, maxround = 5)
Out = GetDEResults(EBOut)

GetMultiFC

GetMultiFC  Calculate the Fold Changes for Multiple Conditions

Description

'GetMultiFC' calculates the Fold Changes for each pair of conditions in a multiple condition study.

Usage

GetMultiFC(EBMultiOut, SmallNum = 0.01)

Arguments

EBMultiOut  The output of EBMultiTest function.
SmallNum    A small number will be added for each transcript in each condition to avoid Inf and NA. Default is 0.01.

Details

Provide the FC (adjusted by the normalization factors) for each pair of comparisons. A small number will be added for each transcript in each condition to avoid Inf and NA. Default is set to be 0.01.

Value

FCMat  The FC of each pair of comparison (adjusted by the normalization factors).
Log2FCMat  The log 2 FC of each pair of comparison (adjusted by the normalization factors).
PostFCMat  The posterior FC of each pair of comparison.
Log2PostFCMat  The log 2 posterior FC of each pair of comparison.
CondMean  The mean of each transcript within each condition (adjusted by the normalization factors).
ConditionOrder  The condition assignment for C1Mean, C2Mean, etc.
Author(s)
Ning Leng

References

See Also
EBMultiTest, PostFC

Examples
```r
data(MultiGeneMat)
MultiGeneMat.small = MultiGeneMat[201:210,]

Conditions = c("C1","C1","C2","C2","C3","C3")

PosParti = GetPatterns(Conditions)
Parti = PosParti[-3,]

MultiSize = MedianNorm(MultiGeneMat.small)

MultiOut = EBMultiTest(MultiGeneMat.small,
NgVector=NULL, Conditions=Conditions,
AllParti=Parti, sizeFactors=MultiSize,
maxround=5)

MultiFC = GetMultiFC(MultiOut)
```

**GetMultiPP**

Posterior Probability of Each Transcript

Description
'GetMultiPP' generates the Posterior Probability of being each pattern of each transcript based on the EBMultiTest output.

Usage
```
GetMultiPP(EBout)
```

Arguments

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBout</td>
<td>The output of EBMultiTest function.</td>
</tr>
</tbody>
</table>
GetNg

Value

PP  The poster probabilities of being each pattern.
MAP  Gives the most likely pattern.
Patterns  The Patterns.

Author(s)

Ning Leng

References


See Also

GetPPMat

Examples

data(MultiGeneMat)
MultiGeneMat.small = MultiGeneMat[201:210,]
Conditions = c("C1","C1","C2","C2","C3","C3")
PosParti = GetPatterns(Conditions)
Parti = PosParti[-3,]
MultiSize = MedianNorm(MultiGeneMat.small)
MultiOut = EBMultiTest(MultiGeneMat.small, NgVector=NULL, Conditions=Conditions, AllParti=Parti, sizeFactors=MultiSize, maxround=5)
MultiPP = GetMultiPP(MultiOut)

---

GetNg  Ng Vector

Description

'GetNg' generates the Ng vector for the isoform level data. (While using the number of isoform in the host gene to define the uncertainty groups.)

Usage

GetNg(IsoformName, GeneName, TrunThre = 3)
GetNg

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsoformName</td>
<td>A vector contains the isoform names.</td>
</tr>
<tr>
<td>GeneName</td>
<td>The gene names of the isoforms in IsoformNames (Should be in the same order).</td>
</tr>
<tr>
<td>TrunThre</td>
<td>The number of uncertainty groups the user wish to define. The default is 3.</td>
</tr>
</tbody>
</table>

Value

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeneNg</td>
<td>The number of isoforms that are contained in each gene.</td>
</tr>
<tr>
<td>GeneNgTrun</td>
<td>The truncated Ng of each gene. (The genes contain more than 3 isoforms are with Ng 3.)</td>
</tr>
<tr>
<td>IsoformNg</td>
<td>The Ng of each isoform.</td>
</tr>
<tr>
<td>IsoformNgTrun</td>
<td>The truncated Ng of each isoform (could be used to define the uncertainty group assignment).</td>
</tr>
</tbody>
</table>

Author(s)

Ning Leng

References


Examples

data(IsoList)

IsoMat = IsoList$IsoMat
IsoNames = IsoList$IsoNames
IsosGeneNames = IsoList$IsosGeneNames
IsoSizes = MedianNorm(IsoMat)
NgList = GetNg(IsoNames, IsosGeneNames)

#IsoNgTrun = NgList$IsoformNgTrun
#IsoEBOut = EBTest(Data = IsoMat, NgVector = IsoNgTrun,
#                   Conditions = as.factor(rep(c("C1","C2"), each=5)),
#                   sizeFactors = IsoSizes, maxround = 5)
GetNormalizedMat

Calculate normalized expression matrix

Description

'GetNormalizedMat' calculates the normalized expression matrix. (Note: this matrix is only used for visualization etc. EBTes and EBMultiTest request *un-adjusted* expressions and normalization factors.)

Usage

GetNormalizedMat(Data, Sizes)

Arguments

Data The data matrix with transcripts in rows and lanes in columns.
Sizes A vector contains the normalization factor for each lane.

Value

The function will return a normalized matrix.

Author(s)

Ning Leng

References


Examples

data(GeneMat)
str(GeneMat)
Sizes = MedianNorm(GeneMat)
NormData = GetNormalizedMat(GeneMat, Sizes)
GetPatterns

*Generate all possible patterns in a multiple condition study*

**Description**

'GetPatterns' generates all possible patterns in a multiple condition study.

**Usage**

```r
GetPatterns(Conditions)
```

**Arguments**

- **Conditions**
  The names of the Conditions in the study.

**Value**

A matrix describe all possible patterns.

**Author(s)**

Ning Leng

**References**


**Examples**

```r
Conditions = c("C1","C1","C2","C2","C3","C3")
PosParti = GetPatterns(Conditions)
```

GetPPMat

*Posterior Probability of Transcripts*

**Description**

'GetPPMat' generates the Posterior Probability of being each pattern of each transcript based on the EBTest output.

**Usage**

```r
GetPPMat(EBout)
```
**GetSelectedPatterns**

**Arguments**

EBout The output of EBTest function.

**Value**

The poster probabilities of being EE (first column) and DE (second column).

**Author(s)**

Ning Leng

**References**


**Examples**

```r
data(GeneMat)
GeneMat.small = GeneMat[c(500:550),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small,
Conditions = as.factor(rep(c("C1","C2"), each=5)),
sizeFactors = Sizes, maxround = 5)
PP = GetPPMat(EBOut)
str(PP)
head(PP)
```

---

**GetSelectedPatterns** Get selected patterns in a multiple condition study

**Description**

'GetSelectedPatterns' get selected patterns in a multiple condition study.

**Usage**

GetSelectedPatterns(EBout)

**Arguments**

EBout Results from EBMultiTest

**Value**

A matrix describe selected patterns.
Author(s)
Ning Leng, Xiuyu Ma

References

Examples
```r
data(MultiGeneMat)
Conditions=c("C1","C1","C2","C2","C3","C3")
MultiSize=MedianNorm(MultiGeneMat)
MultiOut=EBMultiTest(MultiGeneMat,Conditions=Conditions,
sizeFactors=MultiSize)
PosParti=GetSelectedPatterns(MultiOut)
```

IsoList

The simulated data for two condition isoform DE analysis

Description
'IsoList' gives the simulated data for two condition isoform DE analysis.

Usage
data(IsoList)

Source

See Also
GeteMat

Examples
data(IsoList)
**IsoMultiList**  
*The simulated data for multiple condition isoform DE analysis*

**Description**

'IsoMultiList' gives a set of simulated data for multiple condition isoform DE analysis.

**Usage**

```r
data(IsoMultiList)
```

**Source**


**See Also**

IsoList

**Examples**

```r
data(IsoMultiList)
```

---

**Likefun**  
*Likelihood Function of the NB-Beta Model*

**Description**

'Likefun' specifies the Likelihood Function of the NB-Beta Model.

**Usage**

```r
Likefun(ParamPool, InputPool)
```

**Arguments**

- `ParamPool`  
  The parameters that will be estimated in EM.
- `InputPool`  
  The control parameters that will not be estimated in EM.

**Value**

The function will return the log-likelihood.
LikefunMulti

Author(s)
Ning Leng

References

Examples

```R
#x1 = c(.6,.7,.3)
#Input = matrix(rnorm(100,100,1), ncol=10)
#RIn = matrix(rnorm(100,200,1), ncol=10)
#InputPool = list(Input[,1:5], Input[,6:10], Input,
#                 rep(.1,100), 1, RIn, RIn[,1:5], RIn[,6:10], 100)
#Likefun(x1, InputPool)
```

LikefunMulti  
Likelihood Function of the NB-Beta Model In Multiple Condition Test

Description

'LikefunMulti' specifies the Likelihood Function of the NB-Beta Model In Multiple Condition Test.

Usage

LikefunMulti(ParamPool, InputPool)

Arguments

ParamPool  
The parameters that will be estimated in EM.

InputPool  
The control parameters that will not be estimated in EM.

Value

The function will return the log-likelihood.

Author(s)
Ning Leng

References

Examples

```r
#x1 = c(.6,.7,.3)
#Input = matrix(rnorm(100,100,1),ncol=10)
#RIn = matrix(rnorm(100,200,1),ncol=10)
#InputPool = list(list(Input[,1:5],Input[,6:10]),
# Input, cbind(rep(.1, 10), rep(.9,10)), 1,
# RIn, list(RIn[,1:5],RIn[,6:10]),
# 10, rbind(c(1,1),c(1,2)))
#LikefunMulti(x1, InputPool)
```

Description

'LogN' specifies the function to run (one round of) the EM algorithm for the NB-beta model.

Usage

```r
LogN(Input, InputSP, EmpiricalR, EmpiricalRSP, NumOfEachGroup,
AlphaIn, BetaIn, PIn, NoneZeroLength)
```

Arguments

- **Input, InputSP** The expressions among all the samples.
- **NumOfEachGroup** Number of genes in each Ng group.
- **AlphaIn, PIn, BetaIn, EmpiricalR, EmpiricalRSP** The parameters from the last EM step.
- **NoneZeroLength** Number of Ng groups.

Author(s)

Ning Leng

References

Examples

```r
#Input = matrix(rnorm(100,100,1), ncol=10)
#rownames(Input) = paste("g",1:10)
#RIn = matrix(rnorm(100,200,1), ncol=10)
#res = LogN(Input, list(Input[,1:5], Input[,6:10]),
# RIn, list(RIn[,1:5], RIn[,6:10]),
# 10, .6, .7, .3, 1)
```

LogNMulti  

**EM algorithm for the NB-beta model in the multiple condition test**

Description

`'LogNMulti'` specifies the function to run (one round of) the EM algorithm for the NB-beta model in the multiple condition test.

Usage

```r
LogNMulti(Input, InputSP, EmpiricalR, EmpiricalRSP,
           NumOfEachGroup, AlphaIn, BetaIn, PIn,
           NoneZeroLength, AllParti, Conditions)
```

Arguments

- **Input, InputSP**  The expressions among all the samples.
- **NumOfEachGroup**  Number of genes in each Ng group.
- **AlphaIn, PIn, BetaIn, EmpiricalR, EmpiricalRSP**  The parameters from the last EM step.
- **NoneZeroLength**  Number of Ng groups.
- **AllParti**  The patterns of interests.
- **Conditions**  The condition assignment for each sample.

Author(s)

Ning Leng

References

Examples

```
#Input = matrix(rnorm(100,100,1),ncol=10)
#rownames(Input) = paste("g",1:10)
#RIn = matrix(rnorm(100,200,1), ncol=10)
#res = LogNMulti(Input, list(Input[,1:5], Input[,6:10]),
#    RIn, list(RIn[,1:5], RIn[,6:10]), 10, .6, .7,
#    c(.3,.7), 1, rbinding(c(1,1), c(1,2)),
#    as.factor(rep(c("C1","C2"), each=5)))
```

Description

'MedianNorm' specifies the median-by-ratio normalization function from Anders et. al., 2010.

Usage

```
MedianNorm(Data, alternative = FALSE)
```

Arguments

- **Data**: The data matrix with transcripts in rows and lanes in columns.
- **alternative**: if alternative = TRUE, the alternative version of median normalization will be applied. The alternative method is similar to median-by-ratio normalization, but can deal with the cases when all of the genes/isoforms have at least one zero counts (in which case the median-by-ratio normalization will fail).

In more details, in median-by-ratio normalization (denote l_1 as libsize for sample 1 as an example, assume total S samples):

\[
\hat{l}_1 = \frac{\text{median}_g \left( \frac{X_{g1}}{X_{g1}X_{g2}...X_{gS}} \right)^{-S}}{(1)}
\]

which estimates \( l_1 / (l_1 * l_2 * ... * l_S)^{-S} \). Since we have the constrain that \( (l_1 * l_2 * ... * l_S) = 1 \), equation (1) estimates \( l_1 \). Note (1) could also be written as:

\[
\hat{l}_1 = \text{median}_g \left[ \frac{X_{g1}}{X_{g1}X_{g2}...X_{gS}} \right]^{-S}
\]

In the alternative method, we estimate \( l_1/\hat{l}_1, l_1/l_2, ... l_1/l_S \) individually by taking median\(_g(X_{g1}/X_{g1}), \text{median}_g(X_{g1}/X_{g2}) \) ... Then estimate \( l_1 = l_1 / (l_1 * l_2 * ... * l_S)^{-S} \) by taking the geometric mean of these estimates:

\[
\hat{l}_1 = \left[ \text{median}_g(X_{g1}/X_{g1}) \times \text{median}_g(X_{g1}/X_{g2}) \times \text{median}_g(X_{g1}/X_{g3}) \times ... \times \text{median}_g(X_{g1}/X_{gS}) \right]^{-S}
\]

Value

The function will return a vector contains the normalization factor for each lane.
MultiGeneMat

Author(s)
Ning Leng

References

See Also
QuantileNorm

Examples
data(GeneMat)
Sizes = MedianNorm(GeneMat)
#EBOut = EBTest(Data = GeneMat,
# Conditions = as.factor(rep(c("C1","C2"), each=5)),
# sizeFactors = Sizes, maxround = 5)

MultiGeneMat  The simulated data for multiple condition gene DE analysis

Description
'MultiGeneMat' generates a set of the simulated data for multiple condition gene DE analysis.

Usage
data(MultiGeneMat)

Source

See Also
GeneMat

Examples
data(MultiGeneMat)
PlotPattern

Visualize the patterns

Description

'PlotPattern’ generates the visualized patterns before the multiple condition test.

Usage

PlotPattern(Patterns)

Arguments

Patterns  The output of GetPatterns function.

Value

A heatmap to visualize the patterns of interest.

Author(s)

Ning Leng

References


Examples

Conditions = c("C1","C1","C2","C2","C3","C3")
Patterns = GetPatterns(Conditions)
PlotPattern(Patterns)

PlotPostVsRawFC

Plot Posterior FC vs FC

Description

'PlotPostVsRawFC’ helps the users visualize the posterior FC vs FC in a two condition study.

Usage

PlotPostVsRawFC(EBOut, FCOut)
Arguments

EBOut  The output of EBMultiTest function.
FCOut  The output of PostFC function.

Value

A figure shows fold change vs posterior fold change.

Author(s)

Ning Leng

References


See Also

PostFC

Examples

data(GeneMat)
GeneMat.small = GeneMat[c(500:600),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small,
Conditions = as.factor(rep(c("C1","C2"), each=5)),
sizeFactors = Sizes, maxround = 5)
FC = PostFC(EBOut)
PlotPostVsRawFC(EBOut, FC)

---

PolyFitPlot  

'PolyFitPlot' fits the mean-var relationship using polynomial regression.

Usage

PolyFitPlot(X, Y, nterms, xname = "Estimated Mean",
yname = "Estimated Var", pdfname = "", xlim = c(-1,5), ylim = c(-1,7), ChangeXY = F, col = "red")
Arguments

- **X**: The first group of values want to be fitted by the polynomial regression (e.g., Mean of the data).
- **Y**: The second group of values want to be fitted by the polynomial regression (e.g., variance of the data). The length of Y should be the same as the length of X.
- **nterms**: How many polynomial terms want to be used.
- **xname**: Name of the x axis.
- **yname**: Name of the y axis.
- **pdfname**: Name of the plot.
- **xlim**: The x limits of the plot.
- **ylim**: The y limits of the plot.
- **ChangeXY**: If ChangeXY is setted to be TRUE, X will be treated as the dependent variable and Y will be treated as the independent one. Default is FALSE.
- **col**: Color of the fitted line.

Value

The PolyFitPlot function provides a smooth scatter plot of two variables and their best fitting line of polynomial regression.

Author(s)

Ning Leng

References


Examples

data(IsoList)
str(IsoList)
IsoMat = IsoList$IsoMat
IsoNames = IsoList$IsoNames
IsosGeneNames = IsoList$IsosGeneNames
IsoSizes = MedianNorm(IsoMat)
NgList = GetNg(IsoNames, IsosGeneNames)

IsoNgTrun = NgList$IsoformNgTrun
IsoEBOut = EBTest(Data = IsoMat.small,
NgVector = IsoNgTrun,
Conditions = as.factor(rep(c("C1","C2"), each=5)),
sizeFactors = IsoSizes, maxround = 5)

par(mfrow=c(2,2))
#PolyFitValue = vector("list",3)

#for(i in 1:3)
# PolyFitValue[[i]] = PolyFitPlot(IsoEBOut$C1Mean[[i]],
# IsoEBOut$C1EstVar[[i]], 5)

#PolyAll = PolyFitPlot(unlist(IsoEBOut$C1Mean),
# unlist(IsoEBOut$C1EstVar), 5)

#lines(log10(IsoEBOut$C1Mean[[1]])[PolyFitValue[[1]]$sort]),
# PolyFitValue[[1]]$fit[PolyFitValue[[1]]$sort],
# col="yellow", lwd=2)
#lines(log10(IsoEBOut$C1Mean[[2]])[PolyFitValue[[2]]$sort]),
# PolyFitValue[[2]]$fit[PolyFitValue[[2]]$sort],
# col="pink", lwd=2)
#lines(log10(IsoEBOut$C1Mean[[3]])[PolyFitValue[[3]]$sort]),
# PolyFitValue[[3]]$fit[PolyFitValue[[3]]$sort],
# col="green", lwd=2)

#legend("topleft",c("All Isoforms","Ng = 1","Ng = 2","Ng = 3"),
# col = c("red","yellow","pink","green"),
# lty=1, lwd=3, box.lwd=2)

PostFC Calculate the posterior fold change for each transcript across conditions

Description

’PostFC’ calculates the posterior fold change for each transcript across conditions.

Usage

PostFC(EBoutput, SmallNum = 0.01)

Arguments

EBoutput The output from function EBTtest.
SmallNum A small number will be added for each transcript in each condition to avoid Inf and NA. Default is 0.01.

Value

Provide both FC and posterior FC across two conditions. FC is calculated as (MeanC1+SmallNum)/(MeanC2+SmallNum). And Posterior FC is calculated as:

# Post alpha P_a_C1 = alpha + r_C1 * n_C1
# Post beta P_b_C1 = beta + Mean_C1 * n_C1
# \( P_{q\_C1} = \frac{P_{a\_C1}}{P_{a\_C1} + P_{b\_C1}} \)
# Post FC = \( \frac{(1-P_{q\_C1})/P_{q\_c1}}{(1-P_{q\_c2})/P_{q\_c2}} \)

PostFC  The posterior FC across two conditions.
RealFC   The FC across two conditions (adjusted by the normalization factors).
Direction The direction of FC calculation.

Author(s)
Ning Leng

References

See Also
EBTest, GetMultiFC

Examples
```r
data(GeneMat)
GeneMat.small = GeneMat[c(500:550),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small,
               Conditions = as.factor(rep(c("C1","C2"), each=5)),
               sizeFactors = Sizes, maxround = 5)
FC=PostFC(EBOut)
```

QQP The Quantile-Quantile Plot to compare the empirical q’s and simulated q’s from fitted beta distribution

Description
’QQP’ gives the Quantile-Quantile Plot to compare the empirical q’s and simulated q’s from fitted beta distribution.

Usage
```r
QQP(EBOut, GeneLevel = F)
```

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBOut</td>
<td>The output of EBTest or EBMultiTest.</td>
</tr>
<tr>
<td>GeneLevel</td>
<td>Indicate whether the results are from data at gene level.</td>
</tr>
</tbody>
</table>
QuantileNorm

Value
For data with \(n_1\) conditions and \(n_2\) uncertainty groups, \(n_1 \times n_2\) plots will be generated. Each plot represents a subset of the data.

Author(s)
Ning Leng

References

See Also
EBTest, EBMultiTest, DenNHist

Examples
```r
data(GeneMat)
GeneMat.small = GeneMat[c(500:1000),]
Sizes = MedianNorm(GeneMat.small)
EBOut = EBTest(Data = GeneMat.small, Conditions = as.factor(rep(c("C1","C2"), each=5)), sizeFactors = Sizes, maxround = 5)
par(mfrow=c(2,2))
QQP(EBOut)
```

QuantileNorm

Quantile Normalization

Description
'QuantileNorm' gives the quantile normalization.

Usage
QuantileNorm(Data, Quantile)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The data matrix with transcripts in rows and lanes in columns.</td>
</tr>
<tr>
<td>Quantile</td>
<td>The quantile the user wishes to use. Should be a number between 0 and 1.</td>
</tr>
</tbody>
</table>

Details
Use a quantile point to normalize the data.
RankNorm

Value

The function will return a vector contains the normalization factor for each lane.

Author(s)

Ning Leng

References


See Also

MedianNorm

Examples

data(GeneMat)
Sizes = QuantileNorm(GeneMat,.75)
#EBOut = EBTest(Data = GeneMat,
# Conditions = as.factor(rep(c("C1","C2"), each=5)),
# sizeFactors = Sizes, maxround = 5)

RankNorm

Rank Normalization

Description

'RankNorm' gives the rank normalization.

Usage

RankNorm(Data)

Arguments

Data The data matrix with transcripts in rows and lanes in columns.

Value

The function will return a matrix contains the normalization factor for each lane and each transcript.

Author(s)

Ning Leng
See Also

MedianNorm, QuantileNorm

Examples

data(GeneMat)
Sizes = RankNorm(GeneMat)
# Run EBSeq
# EBres = EBTest(Data = GeneData, NgVector = rep(1,10^4),
# Vect5End = rep(1,10^4), Vect3End = rep(1,10^4),
# Conditions = as.factor(rep(c(1,2), each=5)),
# sizeFactors = Sizes, maxround=5)
Index

* DE
  EBMultiTest, 7
  EBTest, 10
  GetDEResults, 15
* FDR
  crit_fun, 5
* Fold Change
  PostFC, 34
* Multiple Condition
  EBMultiTest, 7
* Ng
  GetNg, 19
* Normalization
  GetNormalizedMat, 21
  MedianNorm, 29
  QuantileNorm, 36
  RankNorm, 37
* Posterior Probability
  GetMultiFC, 17
  GetMultiPP, 18
  GetPPMat, 22
  PlotPostVsRawFC, 31
* Q-Q plot
  QQP, 35
* Two condition
  EBTest, 10
  GetDEResults, 15
* beta
  beta.mom, 4
  DenNHist, 6
* datasets
  GeneMat, 14
  IsoList, 24
  IsoMultiList, 25
  MultiGeneMat, 30
* package
  EBSem_NingLeng-package, 3
* patterns
  PlotPattern, 31

beta.mom, 4
crit_fun, 5
DenNHist, 6
EBMultiTest, 7
EBSem_NingLeng
  (EBSem_NingLeng-package), 3
EBSem_NingLeng-package, 3
EBSemTest, 9
EBTest, 10
f0, 12
f1, 13
GeneMat, 14
GetDEResults, 15
GetMultiFC, 17
GetMultiPP, 18
GetNg, 19
GetNormalizedMat, 21
GetPatterns, 22
GetPPMat, 22
GetSelectedPatterns, 23
IsoList, 24
IsoMultiList, 25
Likefun, 25
LikefunMulti, 26
LogN, 27
LogNMulti, 28
MedianNorm, 29
MultiGeneMat, 30
PlotPattern, 31
PlotPostVsRawFC, 31
PolyFitPlot, 32
PostFC, 34
QQP, 35
QuantileNorm, 36
RankNorm, 37