edgeR

November 11, 2009

R topics documented:

`alpha.approxeb` *Estimate the prior weight, alpha*

Description

Estimate the prior weight, using an approximate empirical Bayes rule

Usage

```
alpha.approxeb(object, verbose=TRUE)
```

Arguments

object	DGEList object containing the raw data with elements data (table of counts), group (vector indicating group) and lib.size (vector of library sizes)
verbose	whether to write comments, default true

Value

EBList object with elements sigma2.0.est (numeric scale sigma_0^2 estimate), alpha (estimate for the prior weight, alpha), scores (likelihood scores), infos (Fisher expected information), quantileAdjusted (list from output of quantileAdjust)

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y<-matrix(rnbinom(20,size=1,mu=10),nrow=5)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
alpha<-alpha.approxeb(d)
```

---

approx.expected.info

Approximate of expected information (Fisher information)

Description

Using a linear fit (for simplicity), the expected information from the conditional log likelihood of the dispersion parameter of the negative binomial is calculated over all genes.

Usage

```
approx.expected.info(object, d, qA, robust = FALSE)
```

Arguments

object	DGEList object containing the raw data with elements data (table of counts), group (vector indicating group) and lib.size (vector of library sizes)
d	delta parameter for negative binomial - phi/(phi+1)
qA	list from output of quantileAdjust
robust	logical on whether to use a robust fit, default FALSE

Value

vector of Fisher information approximates (with length same as the number of rows of the original data)

Author(s)

Mark Robinson

Examples

```
set.seed(0)
y<-matrix(rnbinom(40,size=1,mu=10),ncol=4)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
qA<-quantileAdjust(d,alpha=100)
exp.inf<-approx.expected.info(d,1/(1 + qA$r[1]),qA)
```

---

condLogLikDerDelta *Conditional log-likelihood in terms of delta*

Description

Conditional log-likelihood parameterized in terms of delta (`phi / (phi+1)`)

Usage

```
condLogLikDerDelta(y, delta, grid = TRUE, der = 1, doSum = TRUE)
```

Arguments

`y`	matrix with count data (or pseudodata)
`delta`	delta (`phi / (phi+1)`)parameter of negative binomial
`grid`	logical, whether to calculate a grid over the values of delta
`der`	derivative, either 0 (the function), 1 (first derivative) or 2 (second derivative)
`doSum`	logical, whether to sum over samples or not (default `TRUE`

Value

vector or matrix of function/derivative evaluations

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y1<-matrix(rnbinom(10,size=1,mu=10),nrow=5)
v1<-seq(.1,.9,length=9)
ll1<-condLogLikDerDelta(y1,v1,grid=TRUE,der=0,doSum=FALSE)
ll2<-condLogLikDerDelta(y1,delta=.5,grid=FALSE,der=0)
```

condLogLikDerSize *Conditional log-likelihood in terms of size*

Description

Conditional log-likelihood paramterized in terms of size (`1 / phi`)

Usage

```
condLogLikDerSize(y, r, der=1)
```

Arguments

`y`	matrix with count data (or pseudodata)
`r`	size parameter of negative binomial distribution
`der`	derivative, either 0 (the function), 1 (first derivative) or 2 (second derivative)

Value

vector or matrix of function/derivative evaluations

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y1<-matrix(rnbinom(10,size=1,mu=10),nrow=5)
ll2<-condLogLikDerSize(y1,r=10,der=0)
```

deDGEList-class *differential expression of Digital Gene Expression data - class*

Description

A simple list-based class for storing results of differential expression analysis for DGE data

Slots/List Components

Objects of this class contain the following list components: `ps`: list containing estimates of `p` parameter. `r`: numeric vector of size parameter (`1/phi`) where `phi` is negative binomial dispersion. `pseudo`: numeric matrix with the pseudo-counts. `group`: vector giving the experimental group/condition. `M`: numeric scalar with the library size that pseudo counts are mapped to.

Methods

This class inherits directly from class `list` so any operation appropriate for lists will work on objects of this class. `deDGEList` objects also have a `show` method.

Author(s)

Mark Robinson, Davis McCarthy

deDGE *Compute moderated differential expression scores for digital gene expression (DGE) data*

Description

Runs weighted likelihood calculation for moderated estimates of dispersion, and tests for differences in 'tag' abundance between groups

Usage

```
deDGE(object,alpha=500,doPoisson=FALSE,verbose=TRUE)
```

Arguments

object	DGEList containing elements data (matrix: rows-tags, columns-libraries), lib.size, group indicating class
alpha	weight to put on the individual tag's likelihood
doPoisson	logical, whether to fit Poisson model instead of Negative Binomial, default FALSE
verbose	logical, whether to write comments, default TRUE

Value

deDGEList with elements ps (list containing proportion estimates), r (estimates of 1/overdispersion), pseudo (pseudodata generated by quantileAdjust), group (indicating class of each sample), M (geometric mean of library sizes)

Author(s)

Mark Robinson, Davis McCarthy

References

Robinson MD, Smyth GK. 'Small-sample estimation of negative binomial dispersion, with applications to SAGE data.' Biostatistics. 2008 Apr;9(2):321-32.

Robinson MD, Smyth GK. 'Moderated statistical tests for assessing differences in tag abundance.' Bioinformatics. 2007 Nov 1;23(21):2881-7.

Examples

```
# generate raw data from NB, create list object
y<-matrix(rnbinom(20,size=1,mu=10),nrow=5)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))

# find alpha and call main procedure to find differences
alpha<-alpha.approxeb(d)
ms<-deDGE(d,alpha=alpha$alpha)
```

DGEList-class *Digital Gene Expression data - class*

Description

A simple list-based class for storing read counts from digital gene expression technologies.

Slots/List Components

Objects of this class contain the following list components: `data`: numeric matrix containing the read counts. `lib.size`: numeric vector containing the total number of reads for each library (column of `code`). `group`: vector giving the experimental group/condition.

Methods

This class inherits directly from class `list` so any operation appropriate for lists will work on objects of this class. `DGEList` objects also have a `show` method.

Author(s)

Mark Robinson

EBList-class *differential expression of Digital Gene Expression data - class*

Description

A simple list-based class for storing results of the approximate empirical Bayes rule parameters

Slots/List Components

Objects of this class contain the following list components: `sigma2.0.est`: numeric scale `sigma_0^2` estimate. `alpha`: numeric scalar alpha estimate. `scores`: numeric scalar (likelihood) score. `infos`: numeric vector containing the (likelihood) information for each tag. `quantileAdjusted`: list from output of `quantileAdjust`).

Methods

This class inherits directly from class `list` so any operation appropriate for lists will work on objects of this class. `EBList` objects also have a `show` method.

Author(s)

Mark Robinson, Davis McCarthy

`estimatePs` *Estimate expression proportions*

Description

Estimate expression proportions (maximum likelihood with size fixed) based on negative binomial for each tag and sample group (only 2 groups implemented at this point)

Usage

```
estimatePs(object, r, tol = 1e-10, maxit = 30)
```

Arguments

`object`	list containing the raw data with elements `data` (table of counts), `group` (vector indicating group) and `lib.size` (vector of library sizes)
`r`	size parameter of negative binomial
`tol`	tolerance between iterations
`maxit`	maximum number of iterations

Value

list with elements `p.common` (vector giving overall proportion for each tag), `p.group` (matrix with columns giving estimates of proportions for different groups)

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
set.seed(0)
y<-matrix(rnbinom(40,size=1,mu=10),ncol=4)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
ps<-estimatePs(d,r=1)
```

`exactTestNB` *An exact test for differences between two negative binomial groups*

Description

An exact test for differences between two negative binomial groups

Usage

```
exactTestNB(pseudo, group, pair=1:2, mus, r, verbose=TRUE)
```

Arguments

pseudo	data (e.g. quantile adjusted pseudodata) on which to compute Fisher exact statistics
group	group indicator, must be same length as ncol(pseudo)
pair	pair of groups to be compared
mus	vector of means under the null hypothesis (of no difference between groups)
r	preset or estimated negative binomial size parameter. If you want to run a Poisson test, set r very large (e.g. 1000)
verbose	whether to write comments, default TRUE

Value

vector of p-values

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y<-matrix(rnbinom(20,mu=10,size=1.5),nrow=5)
group<-factor(c(1,1,2,2))
mus<-rep(10,5)
f<-exactTestNB(y,group,pair=c(1,2),mus,r=1.5)
```

---

findMaxD2 *Maximizes the negative binomial likelihood*

Description

Maximizes the negative binomial likelihood (a weighted version using the common likelihood given weight alpha) for each tag

Usage

```
findMaxD2(object, alpha = 0.5, grid = TRUE, tol = 1e-05, n.iter = 10, grid.lengt
```

Arguments

object	list containing the raw data with elements data (table of counts), group (vector indicating group) and lib.size (vector of library sizes)
alpha	weight given to common likelihood, set to 0 for individual estimates or large (e.g. 100) for common likelihood
grid	logical, whether to use a grid search (default = TRUE); if FALSE use Newton-Rhapson steps
tol	if grid=FALSE, tolerance for Newton-Rhapson iterations
n.iter	if grid=FALSE, number of Newton-Rhapson iterations
grid.length	length of the grid over which to maximize; default 200

Value

vector of the values of delta that maximize the negative binomial likelihood for each tag (where delta = `phi / (phi+1)` and `phi` is the overdispersion parameter)

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y<-matrix(rnbinom(1000,mu=10,size=2),ncol=4)
d<-DGEList(data=y,group=c(1,1,2,2),lib.size=c(1000:1003))
cml1<-findMaxD2(d,alpha=10)
cml2<-findMaxD2(d,alpha=0)
```

`getData` *Extract data table from DGEList object*

Description

Returns the `data` slot of a DGEList object

Usage

```
getData(object)
```

Arguments

`object`	list containing the raw data with elements `data` (table of counts), `group` (vector indicating group) and `lib.size` (vector of library sizes)

Value

matrix of data (presumably integers)

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
# generate raw data from NB, create list object
y<-matrix(rnbinom(20,size=1,mu=10),nrow=5)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
# should be 5x4
print(dim(getData(d)))
```

interpolateHelper *Quantile Adjustment interpolator*

Description

Helper function to interpolate the quantile function

Usage

```
interpolateHelper(mu, p, r, count.max, verbose=TRUE)
```

Arguments

mu	matrix of means
p	matrix of percentiles
r	scalar, vector or matrix of size parameters
count.max	vector of maximum counts for all tags
verbose	whether to write comments, default true

Value

matrix with quantile-adjusted pseudodata

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y<-matrix(rnbinom(10000,size=2,mu=10),ncol=4)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000,1010),2))
ps<-estimatePs(d,r=2)
N<-prod(d$lib.size)^(1/ncol(d$data))
perc<-pnbinom(d$data-1,size=2,mu=outer(ps$p.common,d$lib.size))+dnbinom(d$data,size=2,mu=
maxcounts<-apply(d$data,1,max)
pseudo<-interpolateHelper(outer(ps$p.common,rep(N,4)),perc,r=2,maxcounts)
```

logLikDerP *Log-likelihood for proportion*

Description

Log-likelihood and derivatives for the proportion parameter of negative binomial (mean = library size * proportion)

Usage

```
logLikDerP(p, y, lib.size, r, der = 0)
```

Arguments

p	vector of proportion parameters to be evaluated
y	matrix of data
lib.size	vector of library sizes
r	size parameter of negative binomial distribution
der	derivative, either 0 (the function), 1 (first derivative) or 2 (second derivative)

Value

vector of evaluations

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
y<-matrix(rnbinom(20,size=1.5,mu=10),nrow=5)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))

this.p<-rowMeans( y/ outer(rep(1,nrow(y)),d$lib.size) )
d1p<-logLikDerP(this.p,y,d$lib.size,r=1.5,der=1)
```

---

plotMA *MA-like plot for deDGEList objects*

Description

Plots M (log-abundance ratio) against A (log-average abundance) for two groups. A smear of points is shown on the left side for those genes with 0 counts in 1 of the 2 classes.

Usage

```
plotMA(object,pair=c(1,2),xlab="A",ylab="M",ylim=NULL,pch=19,eps=0,smearOffset=0
```

Arguments

object	deDGEList object, as output from deDGE
pair	pair of groups to be plotted; default plots groups 1 and 2
xlab	x-axis label
ylab	y-axis label
ylim	limits on y-axis, if left at NULL, scaled to be symmetric about 0
pch	plot character
eps	offset to plot in the log-ratios (i.e. log([p1+eps]/[p2+eps])
smearOffset	offest (to the left of the minimum A value) to plot the smear of 0-in-1-group values
...	further arguments to the plot command

Value

A plot to the current device

Author(s)

Mark Robinson, Davis McCarthy

See Also

`deDGE`

Examples

```
# generate raw data from NB, create list object
y<-matrix(rnbinom(20,size=1,mu=10),nrow=5)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))

# find alpha and call main procedure to find differences
alpha<-alpha.approxeb(d)
ms<-deDGE(d,alpha=alpha$alpha)

# plot it
plotMA(ms)
```

---

quantileAdjust *Normalizes a dataset by using a quantile adjustment*

Description

The function adjusts (you might say normalizes) a dataset, creating pseudodata that represents quantile-adjusted data as if all samples had the same library size, while estimating the dispersion parameter.

Usage

```
quantileAdjust(object, N = prod(object$lib.size)^(1/ncol(object$data)), alpha =
```

Arguments

`object`	list containing the raw data with elements `data` (table of counts), `group` (vector indicating group) and `lib.size` (vector of library sizes)
`N`	library size to normalize to; default is the geometric mean of the original library sizes
`alpha`	weight to put on the individual tag's likelihood
`null.hypothesis`	logical, whether to calculate the means and percentile under the null hypothesis; default is `FALSE`
`n.iter`	number of iterations in estimating the size parameter
`r.init`	initialized value of the size parameter; if `NULL`, then the common value on unadjusted data is used
`tol`	tolerance in estimating the size parameter
`verbose`	whether to write comments, default `true`

Value

list containing several elements used in downstream function calls. `r` is the dispersion estimate, `pseudo` is the quantile-adjusted pseudodata, `ps` is a list containing the abundance estimates, `N` is the common library size and `p` and `mu` are the percentiles and means, respectively that the quantile is based on

Author(s)

Mark Robinson, Davis McCarthy

Examples

```
set.seed(0)
y<-matrix(rnbinom(40,size=1,mu=10),ncol=4)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
qA<-quantileAdjust(d,alpha=100)
```

---

readDGE *Read a list of files containing DGE data*

Description

Reads a list of text files, one for each sample. Files should be tab-delimited with an identifier (could be tag sequence) as the first column and counts as the second column. The function creates one big table with 0s where necessary.

Usage

```
readDGE(files,...)
```

Arguments

`files`	character vector of filenames
`...`	option arguments to send to `read.table`

Value

list with elements `data` (table of counts), `lib.size` (library sizes)

Author(s)

Mark Robinson

Examples

```
#  Read all .txt files from current working directory

## Not run:
files <- dir(pattern="*\\.txt$")
RG <- readDGE(files,sep="\t",header=TRUE,comment.char="",stringsAsFactors=FALSE)
## End(Not run)
```

splitIntoGroupsPseudo *Split pseudodata according to group*

Description

Given a pair of groups, split pseudodata for these groups

Usage

```
splitIntoGroupsPseudo(pseudo, group, pair)
```

Arguments

pseudo	data matrix to be split (e.g. quantile adjusted pseudodata)
group	group indicator, must be same length as ncol(pseudo)
pair	pair of groups to be split from the data

Value

list in which each element is a matrix of count data for an individual group

Author(s)

Davis McCarthy

Examples

```
# generate raw data from NB, create list object
y<-matrix(rnbinom(80,size=1,mu=10),nrow=20)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
rownames(d$data)<-paste("tagno",1:nrow(d$data),sep=".")
z<-splitIntoGroupsPseudo(d$data,d$group,pair=c(1,2))
```

splitIntoGroups *Split the data from a DGEList object according to group*

Description

Split the data from a DGEList object according to group

Usage

```
splitIntoGroups(object)
```

Arguments

object	DGEList, list containing the raw data with elements data (table of counts), group (vector indicating group) and lib.size (vector of library sizes)

Value

list in which each element is a matrix of count data for an individual group

Author(s)

Davis McCarthy

Examples

```
# generate raw data from NB, create list object
y<-matrix(rnbinom(80,size=1,mu=10),nrow=20)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
rownames(d$data)<-paste("tagno",1:nrow(d$data),sep=".")
z<-splitIntoGroups(d)
```

---

tau2.0.objective *Objective function for tau2*

Description

Objective function for tau2 which is used in the rule of how much to squeeze the dispersion parameters towards the common value

Usage

```
tau2.0.objective(tau2.0, info.g, score.g)
```

Arguments

tau2.0	scalar, value for tau2
info.g	observed information for each gene
score.g	observed score (first derivative of log-likelihood) for each gene

Value

scalar, value of objective function at tau2.0

Author(s)

Mark Robinson

Examples

```
y<-matrix(rnbinom(20,size=1,mu=10),nrow=5)
x<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(1000:1001,each=2))
scores <- condLogLikDerDelta(y, delta=0.5, der = 1, doSum = TRUE)
qA <- quantileAdjust(x, alpha = 10, null.hypothesis = TRUE)
exp.inf <- approx.expected.info(x, d=0.5, qA)
sigma2.0.est <- optimize(tau2.0.objective, c(0, 500), info.g = exp.inf, score.g = scores)
```

topTags *Displays the top differentially expressed tags in a table*

Description

Displays/Returns the top DE tags in a data frame

Usage

```
topTags(object,pair,n=10,adj.method= "BH", verbose=TRUE)
```

Arguments

object	deDGEList, output from deDGE
pair	pair of groups to be compared
n	number of tags to display/return
adj.method	method used to adjust P-values, using p.adjust
verbose	whether to write comments, default TRUE

Value

Data frame containing the relative level of expression, log fold changes, unadjusted and adjusted P-values

Author(s)

Mark Robinson, Davis McCarthy

References

Robinson MD, Smyth GK. 'Small-sample estimation of negative binomial dispersion, with applications to SAGE data.' Biostatistics. 2008 Apr;9(2):321-32.

Robinson MD, Smyth GK. 'Moderated statistical tests for assessing differences in tag abundance.' Bioinformatics. 2007 Nov 1;23(21):2881-7.

Examples

```
# generate raw data from NB, create list object
y<-matrix(rnbinom(80,size=1,mu=10),nrow=20)
d<-DGEList(data=y,group=rep(1:2,each=2),lib.size=rep(c(1000:1001),2))
rownames(d$data)<-paste("tagno",1:nrow(d$data),sep=".")

# find alpha and call main procedure to find differences
alpha<-alpha.approxeb(d)
ms<-deDGE(d,alpha=alpha$alpha)

# look at top 10
topTags(ms)
```

`Tu102` *Raw data for several SAGE libraries from the Zhang 1997 Science paper.*

Description

SAGE dataset for 2 tumour samples, 2 normal samples.

Usage

```
data(Tu102)
```

Format

Data frames with 22713, 18794, 16270 and 17703 observations (for Tu102, Tu98, NC2, NC1, respectively) on the following 2 variables.

`Tag_Sequence` a character vector

`Count` a numeric vector

Source

Zhang et al. (1997) *Gene Expression Profiles in Normal and Cancer Cells*. Science, 276, 1268-72.

Index