

Package ‘Polyfit’

April 12, 2018

Type Package

Title Add-on to DESeq to improve p-values and q-values

Version 1.12.0

Date 2014-08-06

Author Conrad Burden

biocViews DifferentialExpression, Sequencing, RNASeq, GeneExpression

Maintainer Conrad Burden <conrad.burden@anu.edu.au>

Depends DESeq

Suggests BiocStyle

Description Polyfit is an add-on to the packages DESeq which ensures the p-value distribution is uniform over the interval [0, 1] for data satisfying the null hypothesis of no differential expression, and uses an adapted Storey-Tibshiran method to calculate q-values.

License GPL (>= 3)

NeedsCompilation no

R topics documented:

Polyfit-package	1
levelPValues	3
pfNbinomTest	4
twoSidedPValueFromDiscrete	5

Index	7
--------------	----------

Polyfit-package	<i>Polyfit add-on to DESeq</i>
-----------------	--------------------------------

Description

implementation the Polyfit add-on to DESeq described in the paper "Improved error estimates for the analysis of differential expression from RNA-seq data"

Details

Package: Polyfit
 Type: Package
 Version: 0.99.3
 Date: 2014-08-06
 License: GPL(>=3)

Polyfit is an add-on to the negative-binomial based packages DESeq for two-class detection of differential expression which ensures the p-value distribution is uniform over the interval [0, 1] for data satisfying the null hypothesis of no differential expression. The first component is the function `pfNbinomTest` which replaces the function `nbinomTest` in DESeq. Its purpose is to smooth point singularities, particularly one at $p = 1$, in the p-value distribution caused by calculating calculating p-values from a discrete distribution. The output from this function should then be passed to the second component, the function `link{levelPValues}`. Its purpose is to apply a variant of the Storey-Tibshirani procedure to shift the p-values so that those corresponding to the null hypothesis have a uniform distribution, and to calculate corresponding q-values (or 'adjusted p-values') for controlling errors via the false discovery rate.

Author(s)

Conrad Burden

Maintainer: conrad.burden@anu.edu.au

References

Burden, C.J., Qureshi, S. and Wilson, S.R. (2014). *Error estimates for the analysis of differential expression from RNA-seq count data*, PeerJ PrePrints 2:e400v1.

Robinson, M., McCarthy, D., and Smyth, G. (2010). *edgeR: a Bioconductor package for differential expression analysis of digital gene expression data*. *Bioinformatics*, **26**, 139-140.

Anders, S. and Huber, W. (2010). *Differential expression analysis for sequence count data*. *Genome Biology*, **11**(10), R106.

Examples

```

# Example using DESeq
cds <- makeExampleCountDataSet()
cds <- estimateSizeFactors( cds )
cds <- estimateDispersions( cds )
nbTPolyfit <- pfNbinomTest( cds, "A", "B" )

lP <- levelPValues(nbTPolyfit$pval)
pvalTab <- cbind(origPval=nbTPolyfit$pval, correctedPval=lP$pValueCorr, qval=lP$qValueCorr)
cat("\n Original and corrected P-values from DESeq \n")
head(pvalTab)

```

levelPValues	<i>Level P-values</i>
--------------	-----------------------

Description

Function to level out a P-value spectrum generated by the Polyfit extension of DESeq by fitting a quadratic function to the right hand portion of the spectrum, produce 'corrected' p-values and q-values using an adapted version of the Storey-Tibsharini procedure

Usage

```
levelPValues(oldPvals, plot = FALSE)
```

Arguments

oldPvals	an array of p-values produced by the Polyfit replacement of the DESeq function <code>pfNbinomTest()</code> or the Plyfit replacement of the edgeR function <code>pfExactTest()</code>
plot	TRUE to plot original and corrected pvalue spectra; FALSE not to plot

Details

`levelPValues` should only be used with P-values generated by the Polyfit function [pfNbinomTest](#), and not with P-values generated by [nbinomTest](#).

Value

List containing

<code>pi0estimate</code>	an estimate of the proportion of genes not differentially expressed
<code>lambdaOptimal</code>	the point in the p-value spectrum past which a quadratic is fitted
<code>pValueCorr</code>	p-values calculated from the levelled spectrum
<code>qValueCorr</code>	q-values calculated from the levelled spectrum
<code>qValueCorrBH</code>	q-values calculated from <code>pValueCorr</code> using Benjamini-Hochberg

Author(s)

Conrad Burden

References

Burden, C.J., Qureshi, S. and Wilson, S.R. (2014). *Error estimates for the analysis of differential expression from RNA-seq count data*, PeerJ PrePrints 2:e400v1.

Examples

```
cds <- makeExampleCountDataSet()
cds <- estimateSizeFactors( cds )
cds <- estimateDispersions( cds )
nbTPolyfit <- pfNbinomTest( cds, "A", "B" )
lP <- levelPValues(nbTPolyfit$pval, plot=TRUE)
pvalTab <- cbind(origPval=nbTPolyfit$pval, correctedPval=lP$pValueCorr)
cat("\n Original and corrected P-values from DESeq \n")
head(pvalTab)
```

pfNbinomTest	<i>The Polyfit extension to the DESeq functions <code>nbinomTest()</code> and <code>nbinomTestForMatrices()</code></i>
--------------	--

Description

Polyfit extensions to the DESeq functions `nbinomTest` and `nbinomTestForMatrices` which test for differences between the base means of two conditions (i.e., for differential expression in the case of RNA-Seq).

Usage

```
pfNbinomTest(cds, condA, condB, pvals_only = FALSE, eps = NULL)
pfNbinomTestForMatrices(countsA, countsB, sizeFactorsA, sizeFactorsB, dispsA, dispsB )
```

Arguments

<code>cds</code>	a <code>CountDataSet</code> with size factors and raw variance functions
<code>condA</code>	one of the conditions in 'cds'
<code>condB</code>	another one of the conditions in 'cds'
<code>pvals_only</code>	return only a vector of (unadjusted) p values instead of the data frame described below
<code>eps</code>	This argument is no longer used. Do not use it
<code>countsA</code>	A matrix of counts, where each column is a replicate
<code>countsB</code>	Another matrix of counts, where each column is a replicate
<code>sizeFactorsA</code>	Size factors for the columns of the matrix 'countsA'
<code>sizeFactorsB</code>	Size factors for the columns of the matrix 'countsB'
<code>dispsA</code>	The dispersions for 'countsA', a vector with one value per gene
<code>dispsB</code>	The same for 'countsB'

Details

These functions have the same behaviour as the DESeq functions `nbinomTest` and `nbinomTestForMatrices`, except that the 'flagpole' in the P-value histogram, particularly at $p = 1$ is redistributed using the function `twoSidedPValueFromDiscrete`.

Value

`pfNbinomTest` gives a data frame with the following columns:

<code>id</code>	The ID of the observable, taken from the row names of the counts slots.
<code>baseMean</code>	The base mean (i.e., mean of the counts divided by the size factors) for the counts for both conditions
<code>baseMeanA</code>	The base mean (i.e., mean of the counts divided by the size factors) for the counts for condition A
<code>baseMeanB</code>	The base mean for condition B
<code>foldChange</code>	The ratio $\text{meanB}/\text{meanA}$

log2FoldChange The log2 of the fold change
 pval The p value for rejecting the null hypothesis 'meanA==meanB'
 padj The adjusted p values (adjusted with 'p.adjust(pval, method="BH")')

pfNbinomTestForMatrices gives a vector of unadjusted p values, one for each row in the counts matrices.

Author(s)

Conrad Burden, conrad.burden@anu.edu.au, based on software by Simon Anders

References

Burden, C.J., Qureshi, S. and Wilson, S.R. (2014). *Error estimates for the analysis of differential expression from RNA-seq count data*, PeerJ PrePrints 2:e400v1.

Anders, S. and Huber, W. (2010). *Differential expression analysis for sequence count data*. Genome Biology, **11**(10), R106.

Examples

```

cds <- makeExampleCountDataSet()
cds <- estimateSizeFactors( cds )
cds <- estimateDispersions( cds )
nbT <- nbinomTest( cds, "A", "B" )
head( nbT )
nbTPolyfit <- pfNbinomTest( cds, "A", "B" )
head( nbTPolyfit )

oldpar <- par(mfrow=c(1,2))
hist(nbT$pval,breaks=seq(0,1,by=0.01),
      xlab="P-value", main="DESeq")
hist(nbTPolyfit$pval,breaks=seq(0,1,by=0.01),
      xlab="P-value", main="polyfit-DESeq")
par(oldpar)

```

twoSidedPValueFromDiscrete

Two sided P-value from discrete distribution

Description

Function to calculate a 2-sided p-value of an observation *xobs* for a finite discrete distribution

$$Prob(X = xobs) = probs[xobs + 1]$$

over the range *xobs* in $(0, 1, \dots, xmax)$ by "squaring off" the distribution to a continuous distribution

Usage

```
twoSidedPValueFromDiscrete(probs, xobs)
```

Arguments

`probs` an array containing the probabilities that X takes the values $0, 1, \dots, x_{max}$
`xobs` a single observed value of X

Details

Note that the returned 2-sided p-value contains a random component, i.e. a given set of input parameters returns a different result each run

Value

A real valued randomised p-value between 0 and 1. If `xobs` is generated with randomly with probability `probs[xobs + 1]` the returned value will be uniformly distributed on the interval $[0, 1]$.

Author(s)

Conrad Burden

Examples

```
pr <- dbinom(0:5, size=5, prob=0.4)
xSample <- rbinom(10000, size=5, prob=0.4)
pvalues <- c()
for(x in xSample){
  pvalues <- c(pvalues, twoSidedPValueFromDiscrete(pr, x))
}
hist(pvalues)
```

Index

- *Topic **\textasciitildekw1**
 - twoSidedPValueFromDiscrete, 5
- *Topic **\textasciitildekw2**
 - twoSidedPValueFromDiscrete, 5
- *Topic **package**
 - Polyfit-package, 1

- levelPValues, 3

- nbinomTest, 2–4
- nbinomTestForMatrices, 4

- pfNbinomTest, 2, 3, 4
- pfNbinomTestForMatrices (pfNbinomTest),
4
- Polyfit (Polyfit-package), 1
- Polyfit-package, 1

- twoSidedPValueFromDiscrete, 4, 5