Package 'Polyfit'

March 24, 2019

Type Package

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

Version 1.16.1

Date 2014-08-06

Author Conrad Burden

biocViews ImmunoOncology, 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 adpated Storey-Tibshiran method to calculate q-values.

License GPL (>= 3)

git_url https://git.bioconductor.org/packages/Polyfit

git_branch RELEASE_3_8

git_last_commit e9c53bc

git_last_commit_date 2019-01-04

Date/Publication 2019-03-23

R topics documented:

`Polyfit-package` *Polyfit add-on to DESeq*

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 unform 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` *Level P-values*

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 pfNbinomTest() or the Plyfit replacement of the edgeR function pfExactTest()
`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

`pi0estimate`	an estimate of the proportion of genes not differentially expressed
`lambdaOptimal`	the point in the p-value spectrum past which a quadratic is fitted
`pValueCorr`	p-values calculated from the levelled spectrum
`qValueCorr`	q-values calculated from the levelled spectrum
`qValueCorrBH`	q-values calculated from pValueCorr 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` *The Polyfit extension to the DESeq functions nbinomTest() and nbinomTestForMatrices()*

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

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

`id`	The ID of the observable, taken from the row names of the counts slots.
`baseMean`	The base mean (i.e., mean of the counts divided by the size factors) for the counts for both conditions
`baseMeanA`	The base mean (i.e., mean of the counts divided by the size factors) for the counts for condition A
`baseMeanB`	The base mean for condition B
`foldChange`	The ratio meanB/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 $(0, 1, ..., xmax)$ 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 $0, 1, ..., xmax$
`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