Package ‘Polyfit’

February 1, 2017

Type Package

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

Version 1.9.0

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 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

<table>
<thead>
<tr>
<th>Package</th>
<th>Polyfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.99.3</td>
</tr>
<tr>
<td>Date</td>
<td>2014-08-06</td>
</tr>
<tr>
<td>License</td>
<td>GPL(&gt;=3)</td>
</tr>
</tbody>
</table>

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 p-values from a discrete distribution. The output from this function should then be passed to the second component, the function `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


### Examples

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

1P <- levelPValues(nbTPolyfit$pval)
pvalTab <- cbind(origPval=nbTPolyfit$pval, correctedPval=1P$pValueCorr, qval=1P$qValueCorr)
cat("n Original and corrected P-values from DESeq \n")
head(pvalTab)
```
levelPValues

**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-Tibshirani procedure

**Usage**

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

**Arguments**

- `oldPvals` an array of p-values produced by the Polyfit replacement of the DESeq function `pfNbinomTest()` or the Polyfit 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**


**Examples**

```r
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)
```
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**

```r
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
twoSidedPValueFromDiscrete

Two sided P-value from discrete distribution

Description

Function to calculate a 2-sided p-value of an observation \(x_{obs}\) for a finite discrete distribution

\[
\text{Prob}(X = x_{obs}) = \text{probs}[x_{obs} + 1]
\]

over the range \(x_{obs}\) in \((0, 1, ..., xmax)\) by "squaring off" the distribution to a continuous distribution

Usage

twoSidedPValueFromDiscrete(probs, xobs)
twoSidedPValueFromDiscrete

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

*Topic \textasciitilde\text{\textsc{kwd1}}
  twoSidedPValueFromDiscrete, 5
*Topic \textasciitilde\text{\textsc{kwd2}}
  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