Package 'DSS'

October 9, 2013

Title Dispersion shrinakge for sequencing data.

Version 1.4.0

Date 2012-12-21

Imports methods

Author Hao Wu <hao.wu@emory.edu>

Depends Biobase, locfdr

Maintainer Hao Wu <hao.wu@emory.edu>

Description DSS is an R library performing the differential expression analysis for RNA-seq count data. DSS implements a new dispersion shrinkage method to estimate the gene-specific biological variance. Extensive simulation results showed that DSS performs favorabily compared to DESeq and edgeR when the variation of biological variances is large.

License GPL

biocViews HighThroughputSequencing, RNAseq, ChIPseq,DifferentialExpression

R topics documented:

DSS-package *Dispersion shrinkage for sequencing data*

Description

DSS is an R library performing the differential expression analysis for RNA-seq count data. Compared with other similar packages (DESeq, edgeR), DSS implements a new dispersion shrinkage method to estimate the gene-specific biological variance. Extensive simulation results showed that DSS performs favorabily compared to DESeq and edgeR when the variation of biological variances is large.

DSS only works for two group comparison at this time. We plan to extend the functionalities and make it work for more general experimental designs in the near future.

Author(s)

Hao Wu <hao.wu@emory.edu>

---

dispersion *Acessor functions for the 'dispersion' slot in a SeqCountData object.*

Description

Dispersion parameter for a gene represents its coefficient of variation of expressions. It characterizes the biological variations.

Usage

```
## S4 method for signature 'SeqCountSet'
dispersion(object)
## S4 replacement method for signature 'SeqCountSet,numeric'
dispersion(object) <- value
```

Arguments

object	A SeqCountData object.
value	A numeric vector with the same length as number of genes.

Details

If the counts from biological replicates are modeled as negative binomial distribution, the variance (v) and mean (m) should hold following relationship: v=m+m^2*phi, where phi is the dispersion. Another interpretation is that phi represents the biological variations among replicates when underlying expressions are modeled as a Gamma distribution.

Author(s)

Hao Wu <hao.wu@emory.edu>

See Also

normalizationFactor

Examples

```
data(seqData)
## obtain
seqData=estNormFactors(seqData, "quantile")
seqData=estDispersion(seqData)
dispersion(seqData)

## assign
dispersion(seqData)=rep(0.1, nrow(exprs(seqData)))
```

---

estDispersion *Estimate and shrink tag-specific dipsersions*

Description

This function first estimate tag-specific dipsersions using a method of moment estimator. Then the dipsersions are shrunk based a penalized likelihood approach.

Usage

```
## S4 method for signature 'SeqCountSet'
estDispersion(seqData, trend=FALSE)
```

Arguments

seqData	An object of SeqCountSet class.
trend	A binary indicator for modeling the dispersion~expression trend.

Details

The function takes and object of seqCountData class and return the same oject with "dispersion" field filled.

With "trend=TRUE" the dependence of dispersion on mean expressions will be modeled. In that case the shrinkage will be performed conditional on mean expressions.

Author(s)

Hao Wu <hao.wu@emory.edu>

Examples

```
data(seqData)
seqData=estNormFactors(seqData)
seqData=estDispersion(seqData)
head(dispersion(seqData))
```

`estNormFactors` *Estimate normalization factors*

Description

This function estimates normalization factors for the input 'seqCountSet' object and return the same object with normalizationFactor field filled or replaced.

Usage

```
## S4 method for signature 'SeqCountSet'
estNormFactors(seqData, method=c("quantile", "total", "median"))
```

Arguments

`seqData`	An object of "SeqCountSet" class.
`method`	Methods to be used in computing normalization factors. Currently available options only include methods to compute normalization factor to adjust for sequencing depths. Available options use (1) "quantile" (default): 75th quantile, (2) "total": total counts, or (3) "median": median counts to constuct the normalization factors. From all methods the normalization factor will be a vector with same length as number of columns for input counts.

Value

The same "SeqCountSet" object with normalizationFactor field filled or replaced.

Author(s)

Hao Wu <hao.wu@emory.edu>

Examples

```
data(seqData)
## compare different methods
seqData=estNormFactors(seqData, "quantile")
k1=normalizationFactor(seqData)
seqData=estNormFactors(seqData, "total")
k2=normalizationFactor(seqData)
cor(k1,k2)

## assign size factor
```

```
normalizationFactor(seqData)=k1

## or normalization factor can be a matrix
dd=exprs(seqData)
f=matrix(runif(length(dd), 1,10), nrow=nrow(dd), ncol=ncol(dd))
normalizationFactor(seqData)=f
head(normalizationFactor(seqData))
```

`normalizationFactor` *Accessor functions for the 'normalizationFactor' slot in a SeqCountData object.*

Description

The normalization factors are used to adjust for technical or biological biases in the sequencing experiments. The factors can either be (1) a vector with length equals to the number of columns of the count data; or (2) a matrix with the same dimension of the count data.

Usage

```
## S4 method for signature 'SeqCountSet'
normalizationFactor(object)
## S4 replacement method for signature 'SeqCountSet,numeric'
normalizationFactor(object) <- value
## S4 replacement method for signature 'SeqCountSet,matrix'
normalizationFactor(object) <- value
```

Arguments

`object`	A SeqCountData object.
`value`	A numeric vector or matrix. If it is a vector it must have length equals to the number of columns of the count data. For matrix it must have the same dimension of the count data.

Details

The vector normalization factors are used mostly to correct for sequencing depth from different datasets. The matrix factor applies a different normalizing constant for each gene at each sample to adjust for a broader range of artifacts such as GC content.

Author(s)

Hao Wu <hao.wu@emory.edu>

See Also

dispersion

Examples

```
data(seqData)
## obtain nomalization factor
seqData=estNormFactors(seqData, "quantile")
normalizationFactor(seqData)

## assign as vector
normalizationFactor(seqData)=rep(1, ncol(exprs(seqData))) ## getan error here

## or assign as a matrix
f=matrix(1, nrow=nrow(exprs(seqData)), ncol=ncol(exprs(seqData)))
normalizationFactor(seqData)=f
```

`SeqCountSet-class` *Class* `"SeqCountSet"` *- container for count data from sequencing experiment*

Description

This class is the main container for storing data from sequencing technology. It is directly inherited fro 'ExpressionSet' class, with two more fields 'normalizationFactor' for normalization factors and 'dispersion' for gene-wise dispersions.

Slots

`normalizationFactor`**:** Normalization factor for counts.

`dispersion`**:** Gene-wise dispersions.

`experimentData`**:** See 'ExpressionSet'.

`assayData`**:** See 'ExpressionSet'.

`phenoData`**:** See 'ExpressionSet'.

`featureData`**:** See 'ExpressionSet'.

`annotation`**:** See 'ExpressionSet'.

`protocolData`**:** See 'ExpressionSet'.

Extends

Class `"ExpressionSet"`, directly. Class `"eSet"`, by class "ExpressionSet", distance 2. Class `"VersionedBiobase"`, by class "ExpressionSet", distance 3. Class `"Versioned"`, by class "ExpressionSet", distance 4.

Constructor

`newSeqCountSet(counts,designs,normalizationFactor,featureData)`: Creates a 'SeqCountSet' object.

`counts` A matrix of integers with rows corresponding to genes and columns for samples.

`designs` A vector representing experimental design. The length of the vector must match the number of columns of input counts. This field can be accessed using 'pData' function.

`normalizationFactor` A vector or matrix of normalization factors for the counts.

`featureData` Additional information for genes as an 'AnnotatedDataFrame' object. This field can be access by using 'featureData' function.

Methods

dispersion, dispersion<- : Access and set gene-wise dispersions.

normalizationFactor, normalizationFactor<- : Access and set normalization factors.

Note

This is similar to 'CountDataSet' in DESeq or 'DGEList' in edgeR.

Author(s)

Hao Wu <hao.wu@emory.edu>

See Also

dispersion, normalizationFactor

Examples

```
counts=matrix(rpois(600, 10), ncol=6)
designs=c(0,0,0,1,1,1)
seqData=newSeqCountSet(counts, designs)
seqData
pData(seqData)
head(exprs(seqData))
```

---

seqData *A simulated 'SeqCountData' object.*

Description

The object is created based on simulation for 1000 genes and two treatment groups with 4 replicates in each group.

Usage

```
data(seqData)
```

Examples

```
data(seqData)
seqData
```

`waldTest` *Perform gene-wise Wald test for two group comparisons for sequencing count data.*

Description

The counts from two groups are modeled as negative binomial random variables with means and dispersions estimated. Wald statistics will be constructed. P-values will be obtained based on Gaussian assumption.

Usage

```
## S4 method for signature 'SeqCountSet'
waldTest(seqData, sampleA, sampleB, equal.var)
```

Arguments

`seqData`	An object of SeqCountSet class.
`sampleA`	The sample labels for the first sample to be compared in two-group comparison.
`sampleB`	The sample labels for the second sample to be compared in two-group comparison.
`equal.var`	A boolean to indicate whether to use the same or different means in two groups for computing variances in Wald test. Default is FALSE.

Details

The input seqCountData object Must have normalizationFactor and dispersion fields filled, e.g., estNormFactors and estDispersion need to be called prior to this. With group means and shrunk dispersions ready, the variances for difference in group means will be constructed based on Negative Binomial distribution. P-values will be obtained under the assumption that the Wald test statistics are normally distributed. Genes with 0 counts in both groups will be assigned 0 for test statistics and 1 for p-values.

Value

A data frame with each row corresponding to a gene. Rows are sorted according to wald test statistics. The columns are:

`gene Index`	index for input gene orders, integers from 1 to the number of genes.
`muA`	sample mean (after normalization) for sample A.
`muB`	sample mean (after normalization) for sample B.
`lfc`	log fold change of expressions between two groups.

difExpr	differences in expressions between two groups.
stats	Wald test statistics.
pval	p-values.
others	input gene annotations supplied as AnnotatedDataFrame when constructed the SeqCountData object.

Author(s)

Hao Wu <hao.wu@emory.edu>

Examples

```
data(seqData)
seqData=estNormFactors(seqData)
seqData=estDispersion(seqData)
result=waldTest(seqData, 0, 1)
head(result)
```

Index