Package ‘SVAPLSseq’

April 15, 2017

Type Package

Title SVAPLSseq-An R package to adjust for the hidden factors of variability in differential gene expression studies based on RNAseq data

Description The package contains functions that are intended for the identification of differentially expressed genes between two groups of samples from RNAseq data after adjusting for various hidden biological and technical factors of variability.

Version 1.0.0

Date 2016-07-13

Author Sutirtha Chakraborty

Maintainer Sutirtha Chakraborty <sutirtha_sutir@yahoo.co.in>

Depends R (>= 3.3)

Imports methods, stats, SummarizedExperiment, edgeR, ggplot2, limma, lmtest, parallel, pls

Suggests BiocStyle

License GPL-3

biocViews GeneExpression, RNASeq, Normalization, BatchEffect

Collate AllClasses.R AllGenerics.R Methods.R svplsSurr.R svplsTest.R

NeedsCompilation no

RoxygenNote 5.0.1

R topics documented:

SVAPLSseq-package *SVAPLSseq: An R package to adjust for the hidden factors of variability in differential gene expression studies based on RNAseq data.*

Description

The package SVAPLSseq contains functions that are intended for the identification and correction of the hidden variability owing to a variety of unknown subject/sample specific and technical effects of residual heterogeneity in an RNAseq gene expression data.

Details

Package:	SVAPLSseq
Type:	Package
License:	GPL-3

The package can be used to find the genes that are truly differentially expressed between two groups of samples from an RNAseq data, after adjusting for different hidden factors of expression heterogeneity. The function svplsSurr operates on the raw data matrix of gene level read counts and extracts the signatures of the underlying hidden variability in the form of a set of surrogate variables. The function svplsTest detects the truly positive genes after correcting for the hidden signals (surrogate variables) extracted by svplsSurr.

Author(s)

Sutirtha Chakraborty.

Maintainer: Sutirtha Chakraborty <statistuta@gmail.com>

References

Boulesteix, A-L. and Strimmer, K. Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. Briefings in Bioinformatics 2007; **8**(1):32–44.

See Also

svplsSurr, svplsTest

Examples

```
##Loading the Simulated Data
data(sim.dat)

## Fitting a linear model with the surrogate variables and detecting the differentially expressed genes
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv <- svplsSurr(dat = sim.dat, group = group)
surr = surr(sv)
fit <- svplsTest(dat = sim.dat, group = group, surr = surr, test = "Wald")
head(sig.genes(fit))
```

prop.vars *Accessor for the 'prop.vars' slot of a 'svplsSurr' object*

Description

Accessor for the 'prop.vars' slot of a 'svplsSurr' object

Usage

```
prop.vars(object)

## S4 method for signature 'svplsSurr'
prop.vars(object)

## S4 method for signature 'svplsSurr'
prop.vars(object)
```

Arguments

`object`	a `svplsSurr` object

Value

A vector of the variance proportions in the data space explained by the significant surrogate variables obtained from `svplsSurr`

Examples

```
data(sim.dat)
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv = svplsSurr(sim.dat, group)
prop.vars(sv)
```

pvs.adj *Accessor for the 'pvs.adj' slot of a 'svplsTest' object*

Description

Accessor for the 'pvs.adj' slot of a 'svplsTest' object

Usage

```
pvs.adj(object)

## S4 method for signature 'svplsTest'
pvs.adj(object)

## S4 method for signature 'svplsTest'
pvs.adj(object)
```

Arguments

object	a svplsTest object

Value

A vector of the FDR-adjusted pvalues for the different genes, obtained from svplsTest.

Examples

```
data(sim.dat)
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv = svplsSurr(sim.dat, group)
surr = surr(sv)
fit = svplsTest(dat = sim.dat, group = group, surr = surr, test = "Wald")
pvs.adj(fit)
```

pvs.unadj *Accessor for the 'pvs.unadj' slot of a 'svplsTest' object*

Description

Accessor for the 'pvs.unadj' slot of a 'svplsTest' object

Usage

```
pvs.unadj(object)

## S4 method for signature 'svplsTest'
pvs.unadj(object)

## S4 method for signature 'svplsTest'
pvs.unadj(object)
```

Arguments

object	a svplsTest object

Value

A vector of the unadjusted pvalues for the different genes, obtained from svplsTest.

Examples

```
data(sim.dat)
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv = svplsSurr(sim.dat, group)
surr = surr(sv)
fit = svplsTest(dat = sim.dat, group = group, surr = surr, test = "Wald")
pvs.unadj(fit)
```

sig.genes — *Accessor for the 'sig.genes' slot of a 'svplsTest' object*

Description

Accessor for the 'sig.genes' slot of a 'svplsTest' object

Usage

```
sig.genes(object)

## S4 method for signature 'svplsTest'
sig.genes(object)

## S4 method for signature 'svplsTest'
sig.genes(object)
```

Arguments

`object`	a `svplsTest` object

Value

A vector of the genes detected to be significantly differentially expressed between the two specified conditions by `svplsTest`.

Examples

```
data(sim.dat)
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv = svplsSurr(sim.dat, group)
surr = surr(sv)
fit = svplsTest(dat = sim.dat, group = group, surr = surr, test = "Wald")
sig.genes(fit)
```

sim.dat — *A simulated RNAseq gene expression count data affected by multiple hidden variables.*

Description

The dataset contains simulated raw RNAseq expression counts corresponding to 1000 genes over 20 subjects distributed equally between two groups 1 and 2. The data is affected by the unknown effects from several technical and sample-specific artefacts. The data has been created to illustrate usage of the functions in this package.

surr *Accessor for the 'surr' slot of a 'svplsSurr' object*

Description

Accessor for the 'surr' slot of a 'svplsSurr' object

Usage

```
surr(object)

## S4 method for signature 'svplsSurr'
surr(object)

## S4 method for signature 'svplsSurr'
surr(object)
```

Arguments

object	a svplsSurr object

Value

A matrix of the significant surrogate variables obtained from svplsSurr

Examples

```
data(sim.dat)
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv = svplsSurr(sim.dat, group)
surr(sv)
```

svplsSurr *svplsSurr*

Description

This function extracts the surrogated estimates of the hidden variables in the data by using the partial least squares (PLS) algorithm on two multivariate random matrices. It provides the user with two options:

(1) **Unsupervised SVAPLS**: Here a standard linear regression model is first used on a transformed version of the expression count matrix to estimate the primary signals of differential expression for all the genes. The fitted model residuals and the transformed count matrix are then organized respectively into two multivariate matrices E and Y, in such a way that each column corresponds to a certain gene. E is then regressed on Y using a Non-linear partial least squares (NPLS) algorithm and the extracted scores in the column-space of Y are deemed as the surrogate variables.

(2) **Supervised SVAPLS**: In case information on a set of control genes (probes) is provided, this function uses a Non-linear partial least squares (NPLS) algorithm to regress Y on a submatrix of Y

(Y.sub) corresponding to the set of controls and scores in the column- space of Y.sub are considered as the surrogate variables.

The function then regresses the first eigenvector of the residual matrix E (for Unsupervised SVAPLS or the control matrix Y.sub for Supervised SVAPLS) on these surrogate variables and tests them for statistical significance with a certain user-specified pvalue cutoff. The variables yielding a pvalue below the cutoff are returned.

Usage

```
svplsSurr(dat, group, controls = NULL, phi = function(x) log(x + const),
  const = 1, pls.method = "oscorespls", max.surrs = 3, cutoff = 10^-7,
  parallel = FALSE, num.cores = NULL, plot = FALSE)
```

Arguments

dat
: A gene expression count matrix or a 'SummarizedExperiment' object or a 'DGE-List' object.

group
: a factor representing the sample indices belonging to the two different groups.

controls
: The set of control probes with no differential expression between the two groups (set to NULL by default).

phi
: The transforming function to be applied on the original gene expression count data (set to be log function with an offset const).

const
: The offset parameter for the transforming function phi (set to 1 by default).

pls.method
: The non-linear partial least squares method to be used. The different options available are: the classical orthogonal scores algorithm ("oscorespls", default), the kernel algorithm ("kernelpls") and wide kernel algorithm ("widekernelpls"). Using the "oscorespls" option is recommended for producing mutually orthogonal surrogate variables.

max.surrs
: The maximum number if surrogate variables to be extracted from the NPLS algorithm (set to 3 by default).

cutoff
: The user-specified pvalue cutoff for testing the significance of the extracted surrogate variables (set to 1e-07 by default).

parallel
: Logical, indicating if the computations should be parallelized or not (set to FALSE by default).

num.cores
: The requested number of cores to be used in the parallel computations inside the function (used only when parallel is TRUE, NULL by default).

plot
: Logical, if TRUE a barplot of the variance proportions explained by the significant surrogate variables is returned (set to FALSE by default).

Value

An svplsSurr object.

Examples

```
##Loading the simulated dataset
data(sim.dat)

##Extracting the significant surrogate variables
group = as.factor(c(rep(1, 10), rep(-1, 10)))
```

```
sv <- svplsSurr(dat = sim.dat, group = group)
print(sv)
```

svplsSurr-class *svplsSurr*

Description

The svplsSurr class. An object of this class contains the following two slots:

surr A data.frame of the significant surrogate variables.

prop.vars A vector of the variance proportions explained by the variables in surr.

svplsTest *svplsTest*

Description

This function incorporates the significant surrogate variables returned by the function svplsSurr in a linear model along with the group variable in order to estimate the group effect more accurately. The reestimated primary signal (group) effects are then used to test the genes for differential expression. The resulting pvalues are further corrected for multiple hypothesis testing at a prespecified FDR level. The significantly differentially expressed genes are finally returned along with their uncorrected and corrected pvalues.

Usage

```
svplsTest(dat, phi = function(x) log(x + const), const = 1, group, surr,
  test = c("Wald", "LRT"), mht.method = "BH", fdr.level = 0.05,
  parallel = FALSE, num.cores = NULL)
```

Arguments

dat A gene expression count matrix or a 'SummarizedExperiment' object or a 'DGEList' object.

phi The transforming function to be applied on the original gene expression count data (set to be log function with an offset const).

const The offset parameter for the transforming function phi (set to 1 by default).

group a factor representing the sample indices belonging to the two different groups.

surr A data.frame of the significant surrogate variables.

test The test to be used for detecting the differentially expressed genes. Options are "Wald" (Wald test with the gene-specific estimated group effects after asjusting for the surrogate variables) and "LRT" (Likelihood Ratio Test).

mht.method The method to be used for the multiple hypothesis correction (set to the Benjamini-Hochberg procedure ("BH") by default).

fdr.level The specified level of the False Discovery Rate (FDR) for the multiple hypothesis testing (set to 0.05 by default).

`parallel` Logical, indicating if the computations should be parallelized or not (set to `FALSE` by default).

`num.cores` The requested number of cores to be used in the parallel computations inside the function (used only when `parallel` is `TRUE`, `NULL` by default).

Value

An `svplsTest` object.

Examples

```
##Loading the simulated dataset
data(sim.dat)

##Fitting a linear model with the surrogate variables and detecting the
##differentially expressed genes
group = as.factor(c(rep(1, 10), rep(-1, 10)))
sv <- svplsSurr(dat = sim.dat, group = group)
surr = surr(sv)
fit = svplsTest(dat = sim.dat, group = group, surr = surr, test = "Wald")

##The detected genes, hidden effect adjusted pvalues, FDR-corrected pvalues and the positive genes detected f
sig.genes(fit)

pvs.unadj(fit)

pvs.adj(fit)
```

`svplsTest-class` *svplsTest*

Description

The `svplsTest` class. An object of this class contains the following three slots:

pvs.unadj The uncorrected pvalues corresponding to the genes after adjusting for the signatures of hidden variability.

pvs.adj The multiple hypothesis corrected pvalues after adjusting for the signatures of hidden variability.

sig.genes The genes detected to be significantly differentially expressed between the two groups.

Index