Package 'tcgsaseq'

May 18, 2016

Type Package

Title Time-Course Gene Set Analysis for RNA-Seq Data

Version 1.0.1

Date 2016-05-14

Depends R (>= 3.0.2)

Imports CompQuadForm, ggplot2, GSA, stats, utils

Suggests limma

Description Gene set analysis of longitudinal RNA-seq data with variance component score test accounting for data heteroscedasticity through precision weights.

LazyData true

License GPL-2 | file LICENSE

BugReports https://github.com/denisagniel/tcgsaseq/issues

RoxygenNote 5.0.1

NeedsCompilation no

Author Denis Agniel [aut],
Boris P. Hejblum [aut, cre]

Maintainer Boris P. Hejblum <bhejblum@hsph.harvard.edu>

Repository CRAN

Date/Publication 2016-05-18 01:23:38

R topics documented:

baduel_small 2
qAbundanceDist 3
sp_weights 4
tcgsaseq 5
tcgsa_seq 6
vc_test_asym 8
vc_test_perm 9
voom_weights 11

Index **13**

baduel_small *Small portion of RNA-seq data from plant physiology study.*

Description

A subsample of the RNA-seq data from Baduel et al. studying Arabidopsis Arenosa physiology.

Usage

```
data(baduel_small)
```

Format

3 objects

- design: a design matrix for the 48 measured samples, containing the following variables:
  - SampleName corresponding column names from expr_norm_corr
  - Population a factor identifying the plant population
  - Age_weeks numeric age of the plant at sampling time (in weeks)
  - Replicate a purely technical variable as replicates are not from the same individual over weeks. Should not be used in analysis.
  - Verbnalized a logical variable indicating whether the plant had undergone vernalization (exposition to cold and short day photoperiods)
- baduel_gmt: a gmt file (see GSA.read.gmt)
- expr_norm_corr: a numeric matrix

Source

http://www.ncbi.nlm.nih.gov/bioproject/PRJNA312410

References

Baduel P, Arnold B, Weisman CM, Hunter B & Bomblies K, Habitat-Associated Life History and Stress-Tolerance Variation in Arabidopsis Arenosa. *Plant Physiology*: 01875, 2015.

Agniel D, Hejblum BP, Variance component score test for time-course gene set analysis of longitudinal RNA-seq data, *submitted*, 2016.

Examples

```
## Not run:
rm(list=ls())
data("baduel_small")
design$Intercept <- 1
design$PopulationKA <- 1*(design$Population=="KA")
design$AgeWeeks_Population <- design$Age_weeks*(design$Population=="KA")
design$Vernalized <- 1*design$Vernalized
design$Vernalized_Population <- design$Vernalized*(design$Population=="KA")
```

```
set.seed(54321)
KAvsTBG <- tcgsa_seq(y=log2(expr_norm_corr+1), x=apply(as.matrix(design[, c("Intercept",
   "Vernalized", "Age_weeks", "Vernalized_Population", "AgeWeeks_Population"), drop=FALSE]),
       2, as.numeric),
                     phi=design[, c("PopulationKA"), drop=FALSE],
                     genesets=baduel_gmt$genesets[c(3,5)],
                     which_test = "permutation", which_weights = "loclin",
                     n_perm=5000, preprocessed = TRUE, doPlot = TRUE)

set.seed(54321)
Cold <- tcgsa_seq(y=log2(expr_norm_corr+1), x=apply(as.matrix(design[, c("Intercept",
   "Age_weeks", "PopulationKA", "AgeWeeks_Population"), drop=FALSE]), 2, as.numeric),
                 phi=design[, c("Vernalized", "Vernalized_Population"), drop=FALSE],
                 genesets=baduel_gmt$genesets[c(3,5)],
                 which_test = "permutation", which_weights = "loclin",
                 n_perm=5000, preprocessed = TRUE, doPlot = TRUE)

## End(Not run)
```

---

qAbundanceDist *Gene abundance proportion distribution of RNA-seq data.*

Description

An example of gene abundance proportion distribution function of RNA-seq data, generated from a real dataset. See supplmenetary material of Law *et al*.

Usage

```
data(qAbundanceDist)
```

Format

A function: `qAbundanceDist`.

Source

http://bioinf.wehi.edu.au/voom/

References

Law CW, Chen Y, Shi W & Smyth GK, voom: Precision weights unlock linear model analysis tools for RNA-seq read counts, *Genome Biology*, 15(2), R29, 2014.

Examples

```
## Not run:
# Get distribution function of abundance proportions
# This distribution was generated from a real dataset
#load(url("http://bioinf.wehi.edu.au/voom/qAbundanceDist.RData"))
data("qAbundanceDist")
curve(qAbundanceDist, from=0, to =0.99)

# Generate baseline proportions for desired number of genes
ngenes <- 10000
baselineprop <- qAbundanceDist( (1:ngenes)/(ngenes+1) )
baselineprop <- baselineprop/sum(baselineprop)

## End(Not run)
```

`sp_weights` *Non parametric local heteroscedasticity weights*

Description

Computes precision weights that account for heteroscedasticity in RNAseq count data based on non-parametric local linear regression estimates.

Usage

```
sp_weights(x, y, phi, preprocessed = FALSE, doPlot = FALSE, bw = c("nrd",
  "ucv", "SJ", "nrd0", "bcv"), kernel = c("gaussian", "epanechnikov",
  "rectangular", "triangular", "biweight", "tricube", "cosine", "optcosine"),
  exact = FALSE)
```

Arguments

`x`	a numeric matrix of size `n x p` containing the model covariates from `n` samples (design matrix).
`y`	anumeric matrix of size `n x G` containing the raw RNAseq counts or preprocessed expression from `n` samples for `G` genes.
`phi`	a numeric design matrix of size `n x K` containing the K basis of time.
`preprocessed`	a logical flag indicating wether the expression data have already been preprocessed (e.g. log2 transformed). Default is `FALSE`, in which case `y` is assumed to contain raw counts and is normalized into log(counts) per million.
`doPlot`	a logical flag indicating wether the mean-variance plot should be drawn. Default is `FALSE`.
`bw`	a character string indicating the smoothing bandwidth selection method to use. See `bandwidth` for details. Possible values are `"ucv"`, `"SJ"`, `"bcv"`, `"nrd"` or `"nrd0"`. Default is `"nrd"`.

kernel	a character string indicating which kernel should be used. Possibilities are "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "tricube", "cosine", "optcosine". Default is "gaussian" (NB: "tricube" kernel corresponds to the loess method).
exact	a logical flag indicating wether the non-parametric weights accounting for the mean-variance relationship should be computed exactly or extrapolated from the interpolation of local regression of the mean against the variance. Default is FALSE, which uses interporlation (faster).

Value

a vector of length n containing the computed precision weights.

See Also

bandwidth density

Examples

```
#rm(list=ls())
set.seed(123)

G <- 10000
n <- 12
p <- 2
y <- sapply(1:G, FUN=function(x){rnbinom(n=n, size=0.07, mu=200)})

x <- sapply(1:p, FUN=function(x){rnorm(n=n, mean=n, sd=1)})
```

---

tcgsaseq *TcGSAseq: a package to perform Time-course Gene Set Analysis of RNA-seq data*

Description

Gene set analysis of longitudinal RNA-seq data with varaince component score test acocunting for data heteroscedasticity through precision weights.

Details

Package:	tcgsaseq
Type:	Package
Version:	1.0.1
Date:	2016-05-14
License:	GPL-2

The main functions of the TcGSAseq package is `tcgsa_seq`

Author(s)

Boris P. Hejblum, Denis Agniel — Maintainer: Boris P. Hejblum

References

Agniel D, Hejblum BP, Variance component score test for time-course gene set analysis of longitudinal RNA-seq data, *submitted*, 2016.

`tcgsa_seq` *Time-course Gene Set Analyis*

Description

Wrapper function for performing gene set analysis of longitudinal RNA-seq data

Usage

```
tcgsa_seq(y, x, phi, genesets, indiv = rep(1, nrow(x)),
  Sigma_xi = diag(ncol(phi)), which_test = c("permutation", "asymptotic"),
  which_weights = c("loclin", "voom"), n_perm = 1000,
  preprocessed = FALSE, doPlot = TRUE, bw = "nrd",
  kernel = c("gaussian", "epanechnikov", "rectangular", "triangular",
  "biweight", "tricube", "cosine", "optcosine"), exact = FALSE,
  padjust_methods = c("BH", "BY", "holm", "hochberg", "hommel", "bonferroni"))
```

Arguments

`y`	a numeric matrix of size `n x G` containing the raw RNA-seq counts or preprocessed expressions from `n` samples for `G` genes.
`x`	a numeric matrix of size `n x p` containing the model covariates from `n` samples (design matrix).
`phi`	a numeric design matrix of size `n x K` containing the `K` variables to be tested
`genesets`	either a vector of index or subscripts that defines which columns of `y` constitute the invesigated geneset. Can also be a `gmt` `list` when several genesets are tested at once.
`indiv`	a vector of length `n` containing the information for attributing each sample to one of the studied individuals. Coerced to be a `factor`.
`Sigma_xi`	a matrix of size `K x K` containing the covariance matrix of the `K` random effects.
`which_test`	a character string indicating which method to use to approximate the variance component score test, either `"permutation"` or `"asymptotic"`. Default is `"permutation"`.

which_weights a character string indicating which method to use to estimate the mean-variance relationship wheights. Possibilities are "loclin", "voom" or NULL (in which case no weighting is performed). Default is "loclin". See sp_weights and voom_weights for details.

n_perm the number of perturbations

preprocessed a logical flag indicating wether the expression data have already been preprocessed (e.g. log2 transformed). Default is FALSE, in which case y is assumed to contain raw counts and is normalized into log(counts) per million.

doPlot a logical flag indicating wether the mean-variance plot should be drawn. Default is FALSE.

bw a character string indicating the smoothing bandwidth selection method to use. See bandwidth for details. Possible values are "ucv", "SJ", "bcv", "nrd" or "nrd0"

kernel a character string indicating which kernel should be used. Possibilities are "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "tricube", "cosine", "optcosine". Default is "gaussian" (NB: "tricube" kernel corresponds to the loess method).

exact a logical flag indicating wether the non-parametric weights accounting for the mean-variance relationship should be computed exactly or extrapolated from the interpolation of local regression of the mean against the variance. Default is FALSE, which uses interporlation (faster).

padjust_methods

multiple testing correction method used if genesets is a list. Default is "BH", i.e. Benjamini-Hochberg procedure for contolling the FDR. Other possibilities are: "holm", "hochberg", "hommel", "bonferroni" or "BY" (for Benjamini-Yekutieli procedure).

Value

A list with the following elements:

- which_test:
- preprocessed:
- n_perm:
- pval: associated p-value

References

Agniel D, Hejblum BP, Variance component score test for time-course gene set analysis of longitudinal RNA-seq data, *submitted*, 2016.

Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, 15(2), R29.

See Also

sp_weights vc_test_perm vc_test_asym p.adjust

`vc_test_asym` *Computes variance component test statistic for longitudinal*

Description

This function computes an approximation of the Variance Component test for a mixture of χ^2s using Davies method from `davies`

Usage

```
vc_test_asym(y, x, indiv = rep(1, nrow(x)), phi, w,
  Sigma_xi = diag(ncol(phi)))
```

Arguments

`y`	a numeric matrix of dim `g x n` containing the raw RNAseq counts for `g` genes from `n` samples.
`x`	a numeric design matrix of dim `n x p` containing the `p` covariates to be adjusted for
`indiv`	a vector of length `n` containing the information for attributing each sample to one of the studied individuals. Coerced to be a `factor`.
`phi`	a numeric design matrix of size `n x K` containing the `K` variables to be tested
`w`	a vector of length `n` containing the weights for the `n` samples.
`Sigma_xi`	a matrix of size `K x K` containing the covariance matrix of the `K` random effects.

Value

A list with the following elements:

- `lam`: TODO
- `q`: TODO
- `q_ext`: TODO
- `score_obs`: approximation of the observed score
- `pval`: associated p-value

See Also

`davies`

Examples

```
#rm(list=ls())
set.seed(123)

##generate some fake data
########################
n <- 100
r <- 12
t <- matrix(rep(1:3), 4, ncol=1, nrow=r)
sigma <- 0.5
b0 <- 1

#under the null:
b1 <- 0
#under the alternative:
b1 <- 0.7
y.tilde <- b0 + b1*t + rnorm(r, sd = sigma)
y <- t(matrix(rnorm(n*r, sd = sqrt(sigma*abs(y.tilde))), ncol=n, nrow=r) +
      matrix(rep(y.tilde, n), ncol=n, nrow=r))
x <- matrix(1, ncol=1, nrow=r)

#run test
asymTestRes <- vc_test_asym(y, x, phi=t, w=matrix(1, ncol=ncol(y), nrow=nrow(y)),
                            Sigma_xi=matrix(1), indiv=rep(1:4, each=3))
asymTestRes$pval
```

`vc_test_perm` *Permutation-based variance component test statistic for longitudinal RNA-seq data*

Description

This function computes an approximation of the Variance Component test for a mixture of χ^2s using permutations. This is preferable to the asymptotic approximation for small sample sizes

Usage

```
vc_test_perm(y, x, indiv = rep(1, nrow(x)), phi, w,
  Sigma_xi = diag(ncol(phi)), n_perm = 1000)
```

Arguments

`y`	a numeric matrix of dim `g x n` containing the raw RNAseq counts for `g` genes from `n` samples.
`x`	a numeric design matrix of dim `n x p` containing the `p` covariates to be adjusted for

indiv	a vector of length n containing the information for attributing each sample to one of the studied individuals. Coerced to be a factor.
phi	a numeric design matrix of size n x K containing the K variables to be tested
w	a vector of length n containing the weights for the n samples.
Sigma_xi	a matrix of size K x K containing the covariance matrix of the K random effects.
n_perm	the number of perturbations

Value

A list with the following elements:

- scores_perm: a vector of length nperm containing the aproximated score test statistics observed for each permutation
- score_obs: approximation of the observed score
- pval: associated p-value

See Also

davies

Examples

```
#rm(list=ls())
set.seed(123)

##generate some fake data
########################
n <- 100
r <- 12
t <- matrix(rep(1:3), 4, ncol=1, nrow=r)
sigma <- 0.5
b0 <- 1

#under the null:
b1 <- 0
#under the alternative:
b1 <- 0.7
y.tilde <- b0 + b1*t + rnorm(r, sd = sigma)
y <- t(matrix(rnorm(n*r, sd = sqrt(sigma*abs(y.tilde))), ncol=n, nrow=r) +
      matrix(rep(y.tilde, n), ncol=n, nrow=r))
x <- matrix(1, ncol=1, nrow=r)

#run test
permTestRes <- vc_test_perm(y, x, phi=t, w=matrix(1, ncol=ncol(y), nrow=nrow(y)),
                            indiv=rep(1:4, each=3), n_perm=100)
permTestRes$pval
```

voom_weights *Precision weights accounting for heteroscedasticity in RNA-seq count data*

Description

Implementation of the procedure descibed in Law *et al* for estimating precision weights from RNA-seq data.

Usage

```
voom_weights(y, x, preprocessed = FALSE, doPlot = FALSE,
  lowess_span = 0.5)
```

Arguments

y	a matrix of size G x n containing the raw RNA-seq counts or preprocessed expressions from n samples for G genes.
x	a matrix of size n x p containing the model covariates from n samples (design matrix).
preprocessed	a logical flag indicating wether the expression data have already been preprocessed (e.g. log2 transformed). Default is FALSE, in which case y is assumed to contain raw counts and is normalized into log(counts) per million.
doPlot	a logical flag indicating wether the mean-variance plot should be drawn. Default is FALSE.
lowess_span	smoother span for the lowess function, between 0 and 1. This gives the proportion of points in the plot which influence the smooth at each value. Larger values give more smoothness.

Value

a vector of length n containing the computed precision weights

References

Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, 15(2), R29.

See Also

lowess approxfun voom

Examples

```
#rm(list=ls())
set.seed(123)

G <- 10000
n <- 12
p <- 2
y <- t(sapply(1:G, FUN=function(x){rnbinom(n=n, size=0.07, mu=200)}))

x <- sapply(1:p, FUN=function(x){rnorm(n=n, mean=n, sd=1)})

my_w <-  voom_weights(y, x, doPlot=TRUE)
w_voom <- limma::voom(counts=y, design=x, plot=TRUE) #slightly faster than us. Same results
all.equal(my_w, w_voom$weights)

## Not run:
microbenchmark::microbenchmark(limma::voom(counts=t(y), design=x, plot=FALSE),
                               voom_weights(x, y, doPlot=FALSE), times=30)
## End(Not run)
```

Index

∗Topic **datasets**
baduel_small, 2
qAbundanceDist, 3

approxfun, *11*

baduel *(*baduel_small*)*, 2
baduel_gmt *(*baduel_small*)*, 2
baduel_small, 2
bandwidth, *4, 5*, *7*

davies, *8*, *10*
density, *5*
design *(*baduel_small*)*, 2

expr_norm_corr *(*baduel_small*)*, 2

gmt, *6*
GSA.read.gmt, *2*

lowess, *11*

p.adjust, *7*

qAbundanceDist, 3

sp_weights, 4, *7*

tcgsa_seq, *6*, 6
tcgsaseq, 5
tcgsaseq-package *(*tcgsaseq*)*, 5

vc_test_asym, *7*, 8
vc_test_perm, *7*, 9
voom, *11*
voom_weights, *7*, 11