Package 'globalSeq'

May 9, 2026

Version 1.41.0

Title Global Test for Counts

Description The method may be conceptualised as a test of overall significance in regression analysis, where the response variable is overdispersed and the number of explanatory variables exceeds the sample size. Useful for testing for association between RNA-Seq and high-dimensional data.

biocViews GeneExpression, ExonArray, DifferentialExpression, GenomeWideAssociation, Transcriptomics, DimensionReduction, Regression, Sequencing, WholeGenome, RNASeq, ExomeSeq, miRNA, MultipleComparison

Depends R (>= 3.0.0)

Suggests knitr, testthat, SummarizedExperiment, S4Vectors

VignetteBuilder knitr

License GPL-3

LazyData true

RoxygenNote 7.0.2

URL https://github.com/rauschenberger/globalSeq

BugReports https://github.com/rauschenberger/globalSeq/issues

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

git_branch devel

git_last_commit 1e1f5dc

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-08

Author Armin Rauschenberger [aut, cre]

Maintainer Armin Rauschenberger <armin.rauschenberger@uni.lu>

Contents

`globalSeq-package` *Negative binomial global test*

Description

Testing for association between RNA-Seq and other genomic data is challenging due to high variability of the former and high dimensionality of the latter.

Using the negative binomial distribution and a random effects model, we developed an omnibus test that overcomes both difficulties. It may be conceptualised as a test of overall significance in regression analysis, where the response variable is overdispersed and the number of explanatory variables exceeds the sample size.

The proposed method can detect genetic and epigenetic alterations that affect gene expression. It can examine complex regulatory mechanisms of gene expression.

Getting started

`omnibus` tests entire covariate sets
`proprius` shows individual contributions
`cursus` analyses the whole genome

The following command opens the vignette:
`utils::vignette("globalSeq")`

More information

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

<a.rauschenberger@vumc.nl>

cursus *Genome-wide analysis*

Description

This function tests for associations between gene expression or exon abundance (Y) and genetic or epigenetic alterations (X). Using the locations of genes (Yloc), and the locations of genetic or epigenetic alterations (Xloc), the expression of each gene is tested for associations with alterations on the same chromosome that are closer to the gene than a given distance (window).

Usage

```
cursus(Y, Yloc, X, Xloc, window,
        Ychr = NULL, Xchr = NULL,
        offset = NULL, group = NULL,
        perm = 1000, nodes = 2,
        phi = NULL, kind = 0.01)
```

Arguments

Y	**RNA-Seq data:** numeric matrix with q rows (genes) and n columns (samples); or a SummarizedExperiment object
Yloc	**location RNA-Seq:** numeric vector of length q (point location)**;** numeric matrix with q rows and two columns (start and end locations)
X	**genomic profile:** numeric matrix with p rows (covariates) and n columns (samples)
Xloc	**location covariates:** numeric vector of length p
window	**maximum distance:** non-negative real number
Ychr	chromosome RNA-Seq**:** factor of length q
Xchr	chromosome covariates**:** factor of length p
offset	numeric vector of length n
group	confounding variable**:** factor of length n
perm	number of iterations**:** positive integer
nodes	number of cluster nodes for parallel computation
phi	dispersion parameters**:** vector of length q
kind	computation **:** number between 0 and 1

Details

Note that Yloc, Xloc and window must be given in the same unit, usually in base pairs. If Yloc indicates interval **locations**, and window is zero, then only covariates between the start and end location of the gene are of interest. Typically window is larger than one million base pairs.

If Y and X include data from a single chromosome, Ychr and Xchr are redundant. If Y or X include data from **multiple chromosomes**, Ychr and Xchr should be specified in order to prevent confusion between chromosomes.

For the simultaneous analysis of **multiple genomic profiles** `X` should be a list of numeric matrices with `n` columns (samples), `Xloc` a list of numeric vectors, and `window` a list of non-negative real numbers. If provided, `Xchr` should be alist of of numeric vectors.

The `offset` is meant to account for different **libary sizes**. By default the `offset` is calculated based on `Y`. Different library sizes can be ignored by setting the `offset` to `rep(1,n)`.

The user can provide the **confounding** variable `group`. Note that each level of `group` must appear at least twice in order to allow stratified permutations.

Efficient alternatives to classical **permutation** (`kind=1`) are the method of control variates (`kind=0`) and permutation in chunks ($0 <$ `kind` < 1) details.

Value

The function returns a dataframe, with the p-values in the first row and the test statistics in the second row.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

RX Menezes, M Boetzer, M Sieswerda, GJB van Ommen, and JM Boer (2009). "Integrated analysis of DNA copy number and gene expression microarray data using gene sets", *BMC Bioinformatics*. 10:203. html pdf (open access)

See Also

The function `omnibus` tests for associations between an overdispersed response variable and a high-dimensional covariate set. The function `proprius` calculates the contributions of individual samples or covariates to the test statistic. All other function of the R package `globalSeq` are `internal`.

Examples

```
# simulate high-dimensional data
n <- 30; q <- 10; p <- 100
Y <- matrix(rnbinom(q*n,mu=10,
    size=1/0.25),nrow=q,ncol=n)
X <- matrix(rnorm(p*n),nrow=p,ncol=n)
Yloc <- seq(0,1,length.out=q)
Xloc <- seq(0,1,length.out=p)
window <- 1

# hypothesis testing
cursus(Y,Yloc,X,Xloc,window)
```

---

intern.chromo *Internal function*

Description

Communicates between `cursus` and `omnibus` by coordinating a chromosome-wide analysis.

Usage

```
intern.chromo(Y, Ystart, Yend, X, Xloc,
              window, offset, group, perm,
              nodes, phi, kind)
```

Arguments

Y	RNA-Seq data: numeric matrix with q rows (genes) and n columns (samples); or a SummarizedExperiment object
Ystart	start location of genes: numeric vector of length q
Yend	end location of genes: NULL or numeric vector of length q
X	genomic profile: numeric matrix with p rows (covariates) and n columns (samples)
Xloc	location covariates: numeric vector of length p
window	maximum distance: non-negative real number
offset	numeric vector of length n
group	confounding variable: factor of length n
perm	number of iterations: positive integer
nodes	number of cluster nodes for parallel computation
phi	dispersion parameters: vector of length q
kind	computation : number between 0 and 1

Value

The function returns a dataframe, with the p-value in the first column, and the test statistic in the second column.

Examples

```
# simulate high-dimensional data
n <- 30
q <- 10
p <- 100
set.seed(1)
Y <- matrix(rnbinom(q*n,mu=10,
    size=1/0.25),nrow=q,ncol=n)
X <- matrix(rnorm(p*n),nrow=p,ncol=n)
Yloc <- seq(0,1,length.out=q)
Xloc <- seq(0,1,length.out=p)
window <- 1

# hypothesis testing
cursus(Y,Yloc,X,Xloc,window)
```

`intern.crude` *Internal function*

Description

Using the parameter estimates `mu` and `phi` and the permutation matrix `perm`, these functions tests for global association between `y` and `X`. The function `intern.crude` calculates p-values by permutation (without repetitions). The functions `intern.focus` and `intern.conva` use different tricks to increase precision and decrease computational expense.

Usage

```
intern.crude(y, X, mu, phi, perm)

intern.focus(y, X, mu, phi, perm, focus)

intern.conva(y, X, mu, phi, perm, offset)
```

Arguments

`y`	response variable: numeric vector of length `n`
`X`	covariate set: numeric matrix with `n` rows (samples) and `p` columns (covariates)
`mu`	mean parameters: numeric vector of length `n`
`phi`	dispersion parameter: non-negative real number
`perm`	permutations: matrix with `n` rows (see example)
`focus`	number between 0 and 1
`offset`	numeric vector of length `n`

Details

The function `intern.focus` uses permutations in chunks. If the remaining permutations do not allow to reach a specified significance level, it stops and rounds the p-value to one.

The function `intern.conva` uses the method of control variates from Senchaudhuri et al. (1995). Roughly speaking, if the test statistics from Rauschenberger et al. (2016) and Goeman et al. (2004) are highly correlated, it returns the asymptotic p-value from Goeman et al. (2004).

Value

Each function returns a dataframe, with the p-value in the first row, and the test statistic in the second row.

References

P Senchaudhuri, CR Mehta, and NR Patel (1995). "Estimating exact p values by the method of control variates or Monte Carlo rescue", *Journal of the American Statistical Association*. 90:640-648 html pdf (restricted access)

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

JJ Goeman, SA van de Geer, F de Kort, and HC van Houwelingen (2004). "A global test for groups of genes: testing association with a clinical outcome", *Bioinformatics*. 20:93-99. html pdf (open access)

See Also

These are `internal` functions. The user functions of the R package `globalSeq` are `cursus`, `omnibus`, and `proprius`.

Examples

```
# simulate high-dimensional data
n <- 30
p <- 100
# set.seed(1)
y <- rnbinom(n,mu=10,size=1/0.25)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)

# prepare arguments
mu <- rep(mean(y),n)
phi <- (var(y)-mu)/mu^2
perm <- intern.permu(n=n,it=99,group=NULL,kind=1)

# perform tests
intern.crude(y,X,mu,phi,perm)
intern.focus(y,X,mu,phi,perm,focus=0.01)
intern.conva(y,X,mu,phi,perm,NULL)
```

`intern.estim` *Internal function*

Description

This functions estimates the parameters of the negative binomial distribution by maximum likelihood. It is called by the functions `omnibus` and `proprius`.

Usage

```
intern.estim(y, offset = NULL)
```

Arguments

`y`	random variable: numeric vector of length `n`
`offset`	numeric vector of length `n`

Details

We assume the negative binomial distribution `y_i ~ NB(mu,phi)`, where the samples are indexed by `i (i=1,...,n)`. Our parametrisation leads to `E[y]= mu` and `Var[y]= mu + phi*mu^2`. With the an offset the model becomes `y_i ~ NB(a_i*mu,phi)`, where the `a_i` are known.

Value

The function returns a list of numeric vectors.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

See Also

This is an `internal` function. The user functions are `cursus`, `omnibus`, and `proprius`.

Examples

```
set.seed(1)
y <- rnbinom(n=1000,mu=10,size=1/0.2)
intern.estim(y)
```

`intern.matrix` *Internal function*

Description

Convert RNA-Seq data to a numeric matrix

Usage

```
intern.matrix(Y)
```

Arguments

`Y`	RNA-Seq data**:** numeric matrix with `q` rows (genes) and `n` columns (samples); or a SummarizedExperiment object

Value

The function returns a matrix.

Examples

```
# simulate RNA-Seq data
Y <- matrix(rnbinom(30,mu=10,size=1/0.2),nrow=10,ncol=3)
rownames(Y) <- paste("gene",1:nrow(Y),sep="")
colnames(Y) <- paste("cell",1:ncol(Y),sep="")

# create data structure
# Z <- SummarizedExperiment::SummarizedExperiment(
#      S4Vectors::SimpleList(counts=Y))

# conversion to matrix
# all.equal(Y,intern.matrix(Z))
```

`intern.permu` *Internal function*

Description

The number of permutations of `n` elements is `n!`. This function randomly rearranges the elements `it` times, and then deletes all duplicates. Thus it finds always less than `it` and `n!` permutations. If a confounding variable is provided, the function uses stratified permutation. This function is called by the functions `omnibus` and `proprius`.

Usage

```
intern.permu(n, it, group, kind)
```

Arguments

`n`	Number of samples.
`it`	Number of repetitions.
`group`	Either `NULL` or a factor of length `n`.
`kind`	computation **:** number between 0 and 1

Value

The function returns a matrix.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

See Also

This is an `internal` function. The user functions are `cursus`, `omnibus`, and `proprius`.

Examples

```
group <- as.factor(c('A','A','B','B','B'))
set.seed(1)
intern.permu(n=5,it=1000,group=group,kind=1)
```

`intern.plot` *Internal function*

Description

This function plots the individual contributions to the test statistic. It is called by the function `proprius`.

Usage

```
intern.plot(u, upper = NULL, xlab = "indices")
```

Arguments

`u`	influence: numeric vector of length `n`
`upper`	critical values: numeric vector of length `n`
`xlab`	label of horizontal axis: character string

Value

The function plots the arguments.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

See Also

This is an `internal` function. The user functions are `cursus`, `omnibus`, and `proprius`.

Examples

```
# simulate influences
set.seed(1)
u <- rchisq(n=100,df=2)

# influence plot
upper <- rep(qchisq(p=0.95,df=2),times=100)
intern.plot(u,upper)
```

`intern.sam` *Internal function*

Description

These functions calculate the contribution of covariate or samples to the test statistic. They are called by the function `proprius`.

Usage

```
intern.sam(y, X, mu, phi)

intern.cov(y, X, mu, phi)
```

Arguments

`y`	response variable: numeric vector of length `n`
`X`	covariate set: numeric matrix with `n` rows (samples) and `p` columns (covariates)
`mu`	mean parameters: numeric vector of length `n`
`phi`	dispersion parameter: non-negative real number

Value

Both functions return a numeric vector.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

JJ Goeman, SA van de Geer, F de Kort, and HC van Houwelingen (2004). "A global test for groups of genes: testing association with a clinical outcome", *Bioinformatics*. 20:93-99. html pdf (open access)

See Also

This is an `internal` function. The user functions of the R package `globalSeq` are `cursus`, `omnibus`, and `proprius`.

Examples

```
# simulate high-dimensional data
n <- 30
p <- 100
set.seed(1)
y <- rnbinom(n,mu=10,size=1/0.25)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)

# prepare arguments
mu <- rep(mean(y),n)
phi <- (var(y)-mean(y))/mean(y)^2
```

```
# decompose test statistic
intern.sam(y,X,mu,phi)
intern.cov(y,X,mu,phi)
```

`intern.score` *Internal function*

Description

This function calculates the test statistic. It is called by the function `omnibus`.

Usage

```
intern.score(y, R, mu, phi)
```

Arguments

`y`	response variable: numeric vector of length `n`
`R`	numeric matrix of dimensions `n*n` (see example)
`mu`	mean parameters**:** numeric vector of length 1 or `n`
`phi`	dispersion parameter**:** non-negative real number

Value

The function returns a real number.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

See Also

This is an `internal` function. The user functions are `cursus`, `omnibus`, and `proprius`.

Examples

```
# simulate high-dimensional data
n <- 30
p <- 100
set.seed(1)
y <- rnbinom(n,mu=10,size=1/0.25)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)

# calculate test statistic
R <- X %*% t(X) / ncol(X)
mu <- mean(y)
phi <- (var(y)-mu)/mu^2
intern.score(y,R,mu,phi)
```

`intern.select` *Internal function*

Description

Communicates between `cursus` and `omnibus` by selecting the covariates of interest.

Usage

```
intern.select(i, Y, Ystart, Yend, X, Xloc,
              window, offset, group,
              perm, phi, kind)
```

Arguments

`i`	index
`Y`	RNA-Seq data**:** numeric matrix with `q` rows (genes) and `n` columns (samples); or a SummarizedExperiment object
`Ystart`	location (or start location)
`Yend`	location (or end location)
`X`	genomic profile**:** numeric matrix with `p` rows (covariates) and `n` columns (samples)
`Xloc`	location covariates**:** numeric vector of length `p`
`window`	maximum distance**:** non-negative real number
`offset`	numeric vector of length `n`
`group`	confounding variable**:** factor of length `n`
`perm`	number of iterations**:** positive integer
`phi`	dispersion parameters**:** vector of length `q`
`kind`	computation **:** number between 0 and 1

Value

The function returns a dataframe, with the p-value in the first column, and the test statistic in the second column.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

See Also

This is an `internal` function. The user functions are `cursus`, `omnibus`, and `proprius`.

Examples

```
# simulate high-dimensional data
n <- 30
q <- 10
p <- 100
set.seed(1)
Y <- matrix(rnbinom(q*n,mu=10,
    size=1/0.25),nrow=q,ncol=n)
X <- matrix(rnorm(p*n),nrow=p,ncol=n)
Yloc <- seq(0,1,length.out=q)
Xloc <- seq(0,1,length.out=p)
window <- 1

# hypothesis testing
cursus(Y,Yloc,X,Xloc,window)
```

`internal` *Internal functions*

Description

This page lists and describes all internal functions of the R package `globalSeq`.

Preparation

`intern.estim` estimates the parameters of the negative binomial distribution by maximum likelihood.
`intern.permu` permutes values across samples, either across all samples or across samples within subgroups.
`intern.score` computes the score test statistic.

Testing

`intern.crude` calculates p-values by permutation.
`intern.focus` calculates p-values by permutation, focusing on a region of interest.
`intern.conva` calculates p-values by permutation, using the method of control variates.

Decomposition

`intern.cov` decomposes the test statistic to show the influence of covariates.
`intern.sam` decomposes the test statistic to show the influence of samples.
`intern.plot` plots the contributions of covariates or samples.

Communication

`intern.chromo` runs through all genes on a chromosome.
`intern.select` identifies local covariates.
`intern.matrix` transforms data to a numeric matrix.

See Also

The user functions of the R package `globalSeq` are `cursus`, `omnibus` and `proprius`.

omnibus *Omnibus test*

Description

Test of association between a count response and one or more covariate sets. This test may be conceptualised as a test of overall significance in regression analysis, where the response variable is overdispersed, and where the number of explanatory variables (p) exceeds the sample size (n). The negative binomial distribution accounts for overdispersion and a random effect model accounts for high dimensionality (p»n).

Usage

```
omnibus(y, X, offset = NULL, group = NULL,
        mu = NULL, phi = NULL,
        perm = 1000, kind = 1)
```

Arguments

y	**response variable:** numeric vector of length n
X	**one covariate set:** numeric matrix with n rows (samples) and p columns (covariates); **multiple covariate sets:** list of numeric matrices with n rows (samples)
offset	numeric vector of length n
group	confounding variable**:** factor of length n
mu	mean parameters**:** numeric vector of length 1 or n
phi	dispersion parameter**:** non-negative real number
perm	number of iterations**:** positive integer
kind	computation **:** number between 0 and 1

Details

The user can provide a common mu for all samples or sample-specific mu, and a common phi. Setting phi equal to zero is equivalent to using the Poisson model. If mu is missing, then mu is estimated from y. If phi is missing, then mu and phi are estimated from y. The offset is only taken into account for estimating mu or phi. By default the offset is rep(1,n).

The user can provide the **confounding** variable group. Note that each level of group must appear at least twice in order to allow stratified permutations.

Efficient alternatives to classical **permutation** (kind=1) are the method of control variates (kind=0) and permutation in chunks (0 < kind < 1) details.

Value

The function returns a dataframe, with the p-value in the first column, and the test statistic in the second column.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

RX Menezes, L Mohammadi, JJ Goeman, and JM Boer (2016). "Analysing multiple types of molecular profiles simultaneously: connecting the needles in the haystack", *BMC Bioinformatics*. 17:77. html pdf (open access)

S le Cessie, and HC van Houwelingen (1995). "Testing the fit of a regression model via score tests in random effects models", *Biometrics*. 51:600-614. html pdf (restricted access)

See Also

The function `proprius` calculates the contributions of individual samples or covariates to the test statistic. The function `cursus` tests for association between RNA-Seq and local genetic or epigenetic alternations across the whole genome. All other functions of the R package `globalSeq` are `internal`.

Examples

```
# simulate high-dimensional data
n <- 30; p <- 100
y <- rnbinom(n,mu=10,size=1/0.25)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)

# hypothesis testing
omnibus(y,X)
```

`proprius` *Decomposition*

Description

Even though the function `omnibus` tests a single hypothesis on a whole covariate set, this function allows to calculate the individual contributions of `n` samples or `p` covariates to the test statistic.

Usage

```
proprius(y, X, type, offset = NULL, group = NULL,
        mu = NULL, phi = NULL,
        alpha = NULL, perm = 1000, plot = TRUE)
```

Arguments

`y`	**response variable:** numeric vector of length `n`
`X`	**covariate set:** numeric matrix with `n` rows (samples) and `p` columns (covariates)
`type`	character '**covariates**' or '**samples**'
`offset`	numeric vector of length `n`
`group`	confounding variable**:** factor of length `n`
`mu`	mean parameters**:** numeric vector of length `1` or `n`

`phi`	dispersion parameter**:** non-negative real number
`alpha`	significance level**:** real number between 0 and 1
`perm`	number of iterations**:** positive integer
`plot`	plot of results**:** logical

Details

The user can provide a common `mu` for all samples or sample-specific `mu`, and a common `phi`. Setting `phi` equal to zero is equivalent to using the Poisson model. If `mu` is missing, then `mu` is estimated from `y`. If `phi` is missing, then `mu` and `phi` are estimated from `y`. The `offset` is only taken into account for estimating `mu` or `phi`.

The user can provide the confounding variable `group`. Note that each level of `group` must appear at least twice in order to allow stratified permutations.

Value

If `alpha=NULL`, then the function returns a numeric vector, and else a list of numeric vectors.

References

A Rauschenberger, MA Jonker, MA van de Wiel, and RX Menezes (2016). "Testing for association between RNA-Seq and high-dimensional data", *BMC Bioinformatics*. 17:118. html pdf (open access)

JJ Goeman, SA van de Geer, F de Kort, and HC van Houwelingen (2004). "A global test for groups of genes: testing association with a clinical outcome", *Bioinformatics*. 20:93-99. html pdf (open access)

See Also

The function `omnibus` tests for associations between an overdispersed response variable and a high-dimensional covariate set. The function `cursus` tests for association between RNA-Seq and local genetic or epigenetic alternations across the whole genome. All other functions of the R package `globalSeq` are `internal`.

Examples

```
# simulate high-dimensional data
n <- 30; p <- 100
y <- rnbinom(n,mu=10,size=1/0.25)
X <- matrix(rnorm(n*p),nrow=n,ncol=p)

# decomposition
proprius(y,X,type="samples")
proprius(y,X,type="covariates")
```

`toydata` *Toydata*

Description

This dataset allows to reproduce the examples shown in the vignette.

Usage

```
data(toydata)
```

Format

A list of numeric vectors and numeric matrices.

Value

All entries are numeric.

Index