

Package ‘globalSeq’

March 17, 2023

Version 1.26.0

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 RELEASE_3_16

git_last_commit ddccaba

git_last_commit_date 2022-11-01

Date/Publication 2023-03-17

Author Armin Rauschenberger [aut, cre]

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

R topics documented:

globalSeq-package	2
cursus	3
omnibus	5
proprius	6
Index	9

globalSeq-package	<i>Negative binomial global test</i>
-------------------	--------------------------------------

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>

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 a list of numeric vectors.

The `offset` is meant to account for different **library 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 functions 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)
```

 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 ($p \gg n$).

Usage

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

Arguments

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

Details

The user can provide a common μ for all samples or sample-specific μ , and a common ϕ . Setting ϕ equal to zero is equivalent to using the Poisson model. If μ is missing, then μ is estimated from y . If ϕ is missing, then μ and ϕ are estimated from y . The `offset` is only taken into account for estimating μ or ϕ . 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 ($0 < \text{kind} < 1$) [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 μ for all samples or sample-specific μ , and a common ϕ . Setting ϕ equal to zero is equivalent to using the Poisson model. If μ is missing, then μ is estimated from y . If ϕ is missing, then μ and ϕ are estimated from y . The `offset` is only taken into account for estimating μ or ϕ .

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 <- rbinom(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")
```


Index

* **documentation**

globalSeq-package, 2

* **methods**

cursus, 3

omnibus, 5

proprius, 6

cursus, 2, 3, 6, 7

details, 4, 5

globalSeq, 4, 6, 7

globalSeq (globalSeq-package), 2

globalSeq-package, 2

internal, 4, 6, 7

omnibus, 2, 4, 5, 6, 7

proprius, 2, 4, 6, 6