

Package ‘rqt’

April 10, 2022

Type Package

Title rqt: utilities for gene-level meta-analysis

Version 1.20.0

Author I. Y. Zhbannikov, K. G. Arbeev, A. I. Yashin.

Maintainer Ilya Y. Zhbannikov <ilya.zhbannikov@duke.edu>

Description Despite the recent advances of modern GWAS methods, it still remains an important problem of addressing calculation an effect size and corresponding p-value for the whole gene rather than for single variant. The R- package rqt offers gene-level GWAS meta-analysis. For more information, see: ``Gene-set association tests for next-generation sequencing data" by Lee et al (2016), Bioinformatics, 32(17), i611-i619, <[doi:10.1093/bioinformatics/btw429](https://doi.org/10.1093/bioinformatics/btw429)>.

URL <https://github.com/izhbannikov/rqt>

BugReports <https://github.com/izhbannikov/rqt/issues>

License GPL

RoxygenNote 6.0.1

Suggests BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

Imports stats,Matrix,ropls,methods,car,RUnit,metap,CompQuadForm,glmnet,utils,pls

Depends R (>= 3.4), SummarizedExperiment

Encoding UTF-8

biocViews GenomeWideAssociation, Regression, Survival, PrincipalComponent, StatisticalMethod, Sequencing

git_url <https://git.bioconductor.org/packages/rqt>

git_branch RELEASE_3_14

git_last_commit 26d52cc

git_last_commit_date 2021-10-26

Date/Publication 2022-04-10

R topics documented:

build.null.model	2
covariates	3
geneTest	3
geneTestMeta	5
genotype	6
get.a	7
phenotype	8
preprocess	8
results	9
rqt	10
rqt-class	11
rqt-general	11
simple.multvar.reg	11
vcov_ridge	12
Index	13

build.null.model	<i>Applies linear of logistic regression to the data.</i>
------------------	---

Description

Applies linear of logistic regression to the data.

Usage

```
build.null.model(y, x, reg.family = "binomial", verbose = FALSE)
```

Arguments

- y A vector with values of dependent variable (outcome).
- x A data.frame of covariates.
- reg.family A regression family. Can be either "binomial" or "gaussian."
- verbose Indicates verbosing output. Default: FALSE.

Value

A list of two: "S" - a dataframe with predictors and "fit" - an object returned by "glm" function.

covariates	<i>This function performs an access to covariates</i>
------------	---

Description

This function performs an access to covariates

An accessor to covariates

Usage

```
covariates(obj)

## S4 method for signature 'rqt'
covariates(obj)
```

Arguments

obj An object of rqt class.

Value

covariates returns the covariates

Examples

```
data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
package="rqt"), header=TRUE))
pheno <- data[,1]
geno <- data[, 2:dim(data)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj <- rqt(pheno=pheno, genotype=geno.obj)
covariates(obj)
```

geneTest	<i>This function performs a gene-level test based on combined effect sizes.</i>
----------	---

Description

This function performs a gene-level test based on combined effect sizes.

geneTest This function performs a gene-level test based on combined effect sizes.

Usage

```
geneTest(obj, ...)

## S4 method for signature 'rqt'
geneTest(obj, perm = 0, STT = 0.2, weight = FALSE,
  cumvar.threshold = 75, out.type = "D", method = "pca",
  scaleData = FALSE, asym.pval = FALSE, penalty = 0.001,
  verbose = FALSE)
```

Arguments

<code>obj</code>	Object of class <code>rqt</code>
<code>...</code>	Additional parameters to pass to the function
<code>perm</code>	Integer indicating the number of permutations to compute p-values. Default: 0.
<code>STT</code>	Numeric indicating soft truncation threshold (STT) to convert to gamma parameter (must be ≤ 0.4). Needed for an optimal parameter a in Gamma-distribution. Default: 0.2. See, for example, Fridley, et al 2013: "Soft truncation thresholding for gene set analysis of RNA-seq data: Application to a vaccine study".
<code>weight</code>	Logical value. Indicates using weights (see Lee et al 2016). Default: FALSE.
<code>cumvar.threshold</code>	Numeric value indicating the explained variance threshold for PCA-like methods. Default: 75
<code>out.type</code>	Character, indicating a type of phenotype. Possible values: D (dichotomous or binary), C (continuous or quantitative).
<code>method</code>	Method used to reduce multicollinearity and account for LD. Default: <code>pca</code> . Another methods available: <code>lasso</code> , <code>ridge</code> , <code>pls</code> .
<code>scaleData</code>	A logic parameter (TRUE/FALSE) indicating scaling of the genotype dataset.
<code>asym.pval</code>	Indicates Monte Carlo approximation for p-values. Default: FALSE.
<code>penalty</code>	A value of penalty parameter for LASSO/ridge regression. Default: 0.001
<code>verbose</code>	Indicates verbosing output. Default: FALSE.

Value

Updated `rqt` object with result slot

Examples

```
data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
  package="rqt"), header=TRUE))
pheno <- data[,1]
geno <- data[, 2:dim(data)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj <- rqt(phenotype=pheno, genotype=geno.obj)
res <- geneTest(obj, method="pca", out.type = "D")
print(res)
```

geneTestMeta	<i>This function performs a gene-level meta-analysis based on combined effect sizes.</i>
--------------	--

Description

This function performs a gene-level meta-analysis based on combined effect sizes.

This function performs a gene-level meta-analysis based on combined effect sizes.

Usage

```
geneTestMeta(objects, ...)

## S4 method for signature 'list'
geneTestMeta(objects, perm = 0, STT = 0.2,
  weight = FALSE, cumvar.threshold = 75, out.type = "D", method = "pca",
  scaleData = FALSE, asym.pval = FALSE, comb.test = "wilkinson",
  penalty = 0.001, verbose = FALSE)
```

Arguments

objects	List of objects of class rqt
...	Additional parameters to pass to the function
perm	Integer indicating the number of permutations to compute p-values. Default: 0.
STT	Numeric indicating soft truncation threshold (STT) to convert to gamma parameter (must be ≤ 0.4). Needed for an optimal parameter a in Gamma-distribution. Default: 0.2. See, for example, Fridley, et al 2013: "Soft truncation thresholding for gene set analysis of RNA-seq data: Application to a vaccine study".
weight	Logical value. Indicates using weights (see Lee et al 2016). Default: FALSE.
cumvar.threshold	Numeric value indicating the explained variance threshold for PCA-like methods. Default: 75
out.type	Character, indicating a type of phenotype. Possible values: D (dichotomous or binary), C (continuous or quantitative).
method	Method used to reduce multicollinearity and account for LD. Default: pca. Other methods available: lasso, ridge, pls.
scaleData	A logic parameter (TRUE/FALSE) indicating scaling of the genotype dataset.
asym.pval	Indicates Monte Carlo approximation for p-values. Default: FALSE.
comb.test	Statistical test for combining p-values.
penalty	Value of penalty parameter for LASSO/ridge regression. Default: 0.001
verbose	Indicates verbosing output. Default: FALSE.

Value

A list of two: (i) final.pvalue - a final p-value across all studies; (ii) pvalueList - p-values for each study;

A list of two: (i) final.pvalue - a final p-value across all studies; (ii) pvalueList - p-values for each study;

Examples

```
data1 <- data.matrix(read.table(system.file("extdata/phengen2.dat",
                                          package="rqt"), skip=1))

pheno <- data1[,1]
geno <- data1[, 2:dim(data1)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj1 <- rqt(pheno=pheno, geno=geno.obj)

data2 <- data.matrix(read.table(system.file("extdata/phengen3.dat",
                                          package="rqt"), skip=1))

pheno <- data2[,1]
geno <- data2[, 2:dim(data2)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj2 <- rqt(pheno=pheno, geno=geno.obj)

data3 <- data.matrix(read.table(system.file("extdata/phengen.dat",
                                          package="rqt"), skip=1))

pheno <- data3[,1]
geno <- data3[, 2:dim(data3)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj3 <- rqt(pheno=pheno, geno=geno.obj)

res.meta <- geneTestMeta(list(obj1, obj2, obj3))
print(res.meta)
```

genotype

This function performs an access to genotype.

Description

This function performs an access to genotype.

A genotype accessor

Usage

```
genotype(obj)

## S4 method for signature 'rqt'
genotype(obj)
```

Arguments

obj An object of rqt class.

Value

genotype returns the genotype

Examples

```
data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
package="rqt"), header=TRUE))
pheno <- data[,1]
geno <- data[, 2:dim(data)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj <- rqt(pheno=pheno, genotype=geno.obj)
genotype(obj)
```

get.a	<i>Get a given STT</i>
-------	------------------------

Description

Get a given STT

Usage

```
get.a(L, STT = 0.2)
```

Arguments

L	TODO
STT	Numeric indicating soft truncation threshold (STT) to convert to gamma parameter (must be ≤ 0.4).

Value

a TODO

phenotype	<i>This function performs an access to phenotype</i>
-----------	--

Description

This function performs an access to phenotype

A phenotype accessor

Usage

```
phenotype(obj)

## S4 method for signature 'rqt'
phenotype(obj)
```

Arguments

obj An object of rqt class.

Value

phenotype returns the phenotype

Examples

```
data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
package="rqt"), header=TRUE))
pheno <- data[,1]
geno <- data[, 2:dim(data)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj <- rqt(phenotype=pheno, genotype=geno.obj)
phenotype(obj)
```

preprocess	<i>Preprocess input data with Principal Component Analysis method (PCA)</i>
------------	---

Description

Preprocess input data with Principal Component Analysis method (PCA)

Usage

```
preprocess(data, pheno = NULL, method = "pca", reg.family = "binomial",
scaleData = FALSE, cumvar.threshold = 75, out.type = "D",
penalty = 0.001, verbose = FALSE)
```

Arguments

<code>data</code>	An input matrix with values of independent variables (predictors).
<code>pheno</code>	A phenotype - column-vector, needed for LASSO/ridge and NULL by default.
<code>method</code>	A dimensionality reduction method. Default: <code>pca</code> .
<code>reg.family</code>	A regression family. Default: <code>"binomial"</code> .
<code>scaledData</code>	A logical variable, indicates wheither or not scaling should be performed. Default: <code>FALSE</code> .
<code>cumvar.threshold</code>	A threshold value for explained variance. Default: <code>75</code>
<code>out.type</code>	An output (phenotype) type. Default: <code>"D"</code>
<code>penalty</code>	Value of penalty parameter for LASSO/ridge regression. Default: <code>0.001</code>
<code>verbose</code>	Indicates verbosing output. Default: <code>FALSE</code> .

Value

A list of one: "S" - a data frame of predictor values.

<code>results</code>	<i>This function performs an access to covariates</i>
----------------------	---

Description

This function performs an access to covariates

An accessor to results

Usage

```
results(obj)
```

```
## S4 method for signature 'rqt'
results(obj)
```

Arguments

<code>obj</code>	An object of <code>rqt</code> class.
------------------	--------------------------------------

Value

`results` returns the results

Examples

```
data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
package="rqt"), header=TRUE))
pheno <- data[,1]
geno <- data[, 2:dim(data)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj <- rqt(pheno=pheno, genotype=geno.obj)
res <- geneTest(obj, method="pca", out.type = "D")
results(res)
```

rqt

*The rqt class constructor***Description**

This function generates rqt class objects

Usage

```
rqt(pheno = NULL, genotype = NULL, covariates = NULL,
    results = NULL)
```

Arguments

pheno	Phenotype (a vector of length N, where N - number of individuals).
genotype	Genotype - an object of class SummarizedExperiment. Should contain one assay (matrix, N by M where N - number of individuals, M - number of genetic variants).
covariates	Covariates, a data frame N by K where N - number of individuals, K - number of covariates
results	A list of two: test statistics: (Q1, Q2, Q3), p-values: (p1.Q1, p2.Q2, p3.Q3)

Value

Object of class rqt

Examples

```
data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
package="rqt"), header=TRUE))
pheno <- data[,1]
geno <- data[, 2:dim(data)[2]]
colnames(geno) <- paste(seq(1, dim(geno)[2]))
geno.obj <- SummarizedExperiment(geno)
obj <- rqt(pheno=pheno, genotype=geno.obj)
print(obj)
```

rqt-class

The rqt class

Description

This class stores parameters and results of the rtq algorithms

Slots

phenotype: Phenotype (a vector of length N, where N - number of individuals).

genotype: Genotype - an object of class SummarizedExperiment. Should contain one assay (matrix, N by M where N - number of individuals, M - number of genetic variants).

covariates: data frame N by K where N - number of individuals, K - number of covariates)

results: A list of two: test statistics (Q1, Q2, Q3), p-values (p1.Q1, p2.Q2, p3.Q3)

rqt-general

General functions of rqt such as accessors and printing.

Description

Common methods for class rqt. This document lists a series of basic methods for the class rqt

Details

Common methods for class rqt

simple.multivar.reg

Applies linear of logistic regression to the data.

Description

Applies linear of logistic regression to the data.

Usage

```
simple.multivar.reg(null.model, Z, verbose = FALSE)
```

Arguments

null.model A fitted null model

Z A genotype matrix

verbose Indicates verbosing output. Default: FALSE.

Value

A list of two: "S" - a dataframe with predictors and "fit" - an object returned by "glm" function.

vcov_ridge	<i>vcov_ridge: returns variance-covariance matrix and standard deviation for ridge/LASSO regression object</i>
------------	--

Description

vcov_ridge: returns variance-covariance matrix and standard deviation for ridge/LASSO regression object

Usage

```
vcov_ridge(x, y, rmod, verbose = FALSE)
```

Arguments

x	Genotype matrix
y	Phenotype
rmod	Ridge/LASSO regression object
verbose	Indicates verbosing output, Default: FALSE.

Value

list(vcov, se). vcov: variance-covariance matrix; se: standard deviation

Index

`build.null.model`, [2](#)

`covariates`, [3](#)
`covariates, rqt-method (covariates)`, [3](#)

`geneTest`, [3](#)
`geneTest, rqt-method (geneTest)`, [3](#)
`geneTestMeta`, [5](#)
`geneTestMeta, list-method (geneTestMeta)`, [5](#)
`genotype`, [6](#)
`genotype, rqt-method (genotype)`, [6](#)
`get.a`, [7](#)

`phenotype`, [8](#)
`phenotype, rqt-method (phenotype)`, [8](#)
`preprocess`, [8](#)

`results`, [9](#)
`results, rqt-method (results)`, [9](#)
`rqt`, [10](#)
`rqt-class`, [11](#)
`rqt-general`, [11](#)

`show.rqt (rqt-general)`, [11](#)
`simple.multivar.reg`, [11](#)

`vcov_ridge`, [12](#)