Package: rqt (via r-universe)

June 10, 2026

Type Package

Title rqt: utilities for gene-level meta-analysis

Version 1.38.0

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>.

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

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

License GPL

RoxygenNote 7.3.2

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

Config/pak/sysreqs cmake libfftw3-dev make libicu-dev libuv1-dev libssl-dev zlib1g-dev

Repository https://bioc-release.r-universe.dev

Date/Publication 2026-04-28 12:45:13 UTC

RemoteUrl https://github.com/bioc/rqt

RemoteRef RELEASE_3_23

RemoteSha 48e32d5d8593b4b68430c8ccec4d4014154fd36f

Contents

`build.null.model` *Applies linear of logistic regregression to the data.*

Description

Applies linear of logistic regregression 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` *This function performs an access to covariates*

Description

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(phenotype=pheno, genotype=geno.obj)
covariates(obj)
```

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

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

obj	Object 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 (continous or quantitative).
method	Method used to reduce multicollinerity and account for LD. Default: pca. Another 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.
penalty	A value of penalty parameter for LASSO/ridge regression. Default: 0.001
verbose	Indicates verbosing output. Default: FALSE.

Value

Updated rqt object with result slot

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(phenotype=pheno, genotype=geno.obj)
res <- geneTest(obj, method="pca", out.type = "D")
print(res)
```

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

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 (continous or quantitative).
method	Method used to reduce multicollinerity and account for LD. Default: pca. Another 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(phenotype=pheno, genotype=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(phenotype=pheno, genotype=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(phenotype=pheno, genotype=geno.obj)

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

---

geneTestOne *get.reg.family*

Description

get.reg.family

Usage

```
geneTestOne(
  phenotype,
  genotype,
  covariates,
  STT = 0.2,
  weight = FALSE,
  cumvar.threshold = 75,
  method = "pca",
  out.type = "D",
  scaleData = FALSE,
  penalty = 0.001,
  verbose = FALSE
)
```

Arguments

phenotype	phenotype
genotype	genotype
covariates	covariates
STT	STT
weight	weight
cumvar.threshold	cumvar.threshold
method	method
out.type	out.type
scaleData	scaleData
penalty	penalty
verbose	verbose

Value

rslt

---

genotype *This function performs an access to genotype.*

Description

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(phenotype=pheno, genotype=geno.obj)
genotype(obj)
```

---

get.a *Get a given STT*

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 number from gamma distribution

---

get.reg.family *get.reg.family*

Description

get.reg.family

Usage

```
get.reg.family(out.type = "D")
```

Arguments

out.type	out.type

Value

reg.family

---

phenotype *This function performs an access to phenotype*

Description

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 *Preprocess input data with Principal Component Analysis method (PCA)*

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

data	An input matrix with values of independent variables (predictors).
pheno	A phenotype - column-vector, needed for LASSO/ridge and NULL by default.
method	A dimensionality reduction method. Default: pca.
reg.family	A regression family. Default: "binomial".
scaleData	A logical variable, indicates wheither or not scaling should be performed. Default: FALSE.
cumvar.threshold	A threshold value for explained variance. Default: 75

out.type	An output (phenotype) type. Default: "D"
penalty	Value of penalty parameter for LASSO/ridge regression. Default: 0.001
verbose	Indicates verbosing output. Default: FALSE.

Value

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

`preprocessLASSO` *preprocessLASSO*

Description

preprocessLASSO

Usage

```
preprocessLASSO(data, pheno, reg.family, penalty = 0.001)
```

Arguments

data	data
pheno	pheno data
reg.family	reg.family
penalty	penalty Default: FALSE.

Value

list(S, fit, model)

`preprocessPCA` *preprocessPCA*

Description

preprocessPCA

Usage

```
preprocessPCA(data, scaleData, cumvar.threshold, verbose)
```

Arguments

data	data
scaleData	scaled data
cumvar.threshold	cumvar.threshold
verbose	Indicates verbosing output Default: FALSE.

Value

list(S, indices, model).

---

preprocessPLS *preprocessPLS*

Description

preprocessPLS

Usage

```
preprocessPLS(data, pheno, scaleData, cumvar.threshold, out.type)
```

Arguments

data	data
pheno	pheno data
scaleData	scaleData
cumvar.threshold	cumvar.threshold
out.type	out.type Default: FALSE.

Value

list(S, Y, model)

preprocessRidge *preprocessLASSO*

Description

preprocessLASSO

Usage

```
preprocessRidge(data, pheno, reg.family, penalty = 0.001)
```

Arguments

data	data
pheno	pheno data
reg.family	reg.family
penalty	penalty Default: FALSE.

Value

list(S, fit, model)

---

results *This function performs an access to covariates*

Description

An accessor to results

Usage

```
results(obj)

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

Arguments

obj	An object of rqt 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(phenotype=pheno, genotype=geno.obj)
res <- geneTest(obj, method="pca", out.type = "D")
results(res)
```

`ridge_se` *Importing required packages and functions*

Description

Importing required packages and functions

Usage

```
ridge_se(xs, y, yhat, my_mod, verbose = FALSE)
```

Arguments

`xs`	Genotype matrix
`y`	Phenotype
`yhat`	Ridge/LASSO regression object
`my_mod`	Ridge/LASSO regression object
`verbose`	Indicates verbosing output, Default: FALSE.

Value

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

---

`rqt` *The rqt class constructor*

Description

This function generates rqt class objects

Usage

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

Arguments

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	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(phenotype=pheno, genotype=geno.obj)
print(obj)
```

---

rqt-class *The rqt class*

Description

This class stores parameters and results of the rtq algorithms

Value

None

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

Value

None

simple.multvar.reg *Applies linear of logistic regregression to the data.*

Description

Applies linear of logistic regregression to the data.

Usage

```
simple.multvar.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` *vcov_ridge: returns variance-covariance matrix and standard deviation for ridge/LASSO regression object*

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