Package: loci2path (via r-universe)

November 4, 2026

Type Package

Title Loci2path: regulatory annotation of genomic intervals based on tissue-specific expression QTLs

Version 1.32.0

Author Tianlei Xu

Maintainer Tianlei Xu <tianlei.xu@emory.edu>

Description loci2path performs statistics-rigorous enrichment analysis of eQTLs in genomic regions of interest. Using eQTL collections provided by the Genotype-Tissue Expression (GTEx) project and pathway collections from MSigDB.

License Artistic-2.0

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports pheatmap, wordcloud, RColorBrewer, data.table, methods, grDevices, stats, graphics, GenomicRanges, BiocParallel, S4Vectors

Suggests BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

biocViews FunctionalGenomics, Genetics, GeneSetEnrichment, Software, GeneExpression, Sequencing, Coverage, BioCarta

URL https://github.com/StanleyXu/loci2path

BugReports https://github.com/StanleyXu/loci2path/issues

Collate 'allClasses.R' 'allGenerics.R' 'allAccessor.R' 'check.geneid.R' 'loci2path.demo-data.R' 'loci2pathResult-methods.R' 'query-methods.R' 'show-methods.R'

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:46:59 UTC

RemoteUrl https://github.com/bioc/loci2path

RemoteRef RELEASE_3_23

RemoteSha 97dcb4d844cce8db307e69d3dd63c58169e03ac6

Contents

biocarta . . . 2
check.geneid . . . 3
eqtlSet-class . . . 3
eset.list . . . 4
geneSet-class . . . 5
getHeatmap . . . 6
getMat . . . 7
getPathDescription . . . 8
getPval . . . 9
getTissueDegree . . . 10
getWordcloud . . . 11
loci2pathResult-class . . . 11
query . . . 12
query.gr . . . 14

Index **16**

`biocarta` *eQTL geneset enrichment query demo data*

Description

Demo Data to show how to perform eQTL-geneset enrichment query.

Usage

```
data(loci2path.demo)
```

Format

An object of class `geneSet` of length 1.

Details

`eqtl.set.list` A list of eQTLset objects; eQTL data are collected from GTeX.

`biocarta` A Geneset object; Geneset are from Broad Institute's MSigDB: curated gene set, category 'cp': BIOCARTA

`query.gr` A Genomic Region object; Query regions are from immunoBase, crohn's disease.

Examples

```
data(loci2path.demo)
```

check.geneid *check compatibility of gene identifiers between eQTL set and gene set*

Description

This function perform enrichment test between one eQTL set and one gene set

Usage

```
check.geneid(e.set, g.set)
```

Arguments

e.set	an eqtlSet object; the eQTL set to be queried against
g.set	an object of geneSet class; the gene set to be tested

Value

a data.frame shows the number of genes from (1) eqtl Set (2) gene Set (3) shared

Examples

```
check.geneid(eset.list$Skin, biocarta)
```

eqtlSet-class *eqtlSet Class*

Description

eqtlSet Class contains information for eqtl-gene association, gene identifier, position of SNPs, etc.

Usage

```
tissue(x)

eqtlId(x)

eqtlRange(x)

eqtlGene(x)

## S4 method for signature 'eqtlSet'
tissue(x)

## S4 method for signature 'eqtlSet'
eqtlId(x)
```

```
## S4 method for signature 'eqtlSet'
eqtlRange(x)

## S4 method for signature 'eqtlSet'
eqtlGene(x)
```

Arguments

x	An eqtlSet object

Value

Object of class eqtlSet

Slots

`tissue` character; name of the cell/tissue of the eQTL study

`eqtlId` character; name of the SNPs

`eqtlRange` GenomicRanges; position of the SNPs

`gene` character; gene identifier

Examples

```
require(GenomicRanges)
brain.file <- system.file("extdata", "eqtl/brain.gtex.txt",
    package="loci2path")
tab <- read.table(brain.file, stringsAsFactors=FALSE, header=TRUE)
eqtlRange <- GRanges(seqnames=Rle(tab$snp.chr),
    ranges=IRanges(start=tab$snp.pos,
    width=1))
brain.eset <- eqtlSet(tissue="brain",
    eqtlId=tab$snp.id,
    eqtlRange=eqtlRange,
    gene=as.character(tab$gene.entrez.id))
tissue(brain.eset)
head(eqtlId(brain.eset))
eqtlRange(brain.eset)
head(eqtlGene(brain.eset))
```

---

eset.list *eQTL geneset enrichment query demo data*

Description

Demo Data to show how to perform eQTL-geneset enrichment query.

Usage

```
data(loci2path.demo)
```

Format

An object of class `list` of length 3.

Details

`eset.list` A list of eQTLset objects; eQTL data are collected from GTeX.

`biocarta` A Geneset object; Geneset are from Broad Institute's MSigDB: curated gene set, category 'cp': BIOCARTA

`query.gr` A Genomic Region object; Query regions are from immunoBase, Psoriasis disease.

Examples

```
data(loci2path.demo)
```

---

geneSet-class *geneSet Class*

Description

geneSet Class contains information for names of gene sets and a list of gene sets

Usage

```
numGene(x)

description(x)

geneSetList(x)

## S4 method for signature 'geneSet'
numGene(x)

## S4 method for signature 'geneSet'
description(x)

## S4 method for signature 'geneSet'
geneSetList(x)
```

Arguments

x	An geneSet object

Value

Object of class geneSet

Slots

`numGene` numeric; the total number of all genes; This number is used in enrichment tests

`description` vector of character; additional information for gene sets, such as names, URLs, a short description, etc.

`geneSetList` list; a list of gene sets; each member is a vector containing a group of gene identifiers

Examples

```
biocarta.link.file <- system.file("extdata",
    "geneSet/biocarta.txt", package="loci2path")
biocarta.link <- read.delim(biocarta.link.file, header=FALSE,
    stringsAsFactors=FALSE)
biocarta.set.file <- system.file("extdata", "geneSet/biocarta.set.txt",
    package="loci2path")
set.geneid <- read.table(biocarta.set.file, stringsAsFactors=FALSE)
set.geneid <- strsplit(set.geneid[,1], split=",")
names(set.geneid) <- biocarta.link[,1]
biocarta <- geneSet(
    geneSetList=set.geneid,
    description=biocarta.link[,2],
    numGene=31847)
numGene(biocarta)
head(description(biocarta))
head(geneSetList(biocarta))
```

---

getHeatmap *Generate heatmap of enrichment matrix from query result*

Description

This function generate the enrichment heatmap using pheatmap package.

Usage

```
getHeatmap(res, ...)

## S4 method for signature 'loci2pathResult'
getHeatmap(res, main = "",
  test.method = c("gene", "eqtl", "glm"), filter.quantile = 0.5,
  max.ptw.gene = 5000)
```

Arguments

res	query result from function query.egset.list()
...	additional params
main	title of the heatmap, default is ""
test.method	Choose which enrichment test should be used to retrive p-values from. Options include:"gene"(default, gene-based fisher's exact test),"eqtl" (eqtl based fisher's exact test), "glm" (ordered query)
filter.quantile	Filter option; choose the max quantile of all p-values being kept in the matrix; default is 0.5, which means p-values larger than median p-values are discarded
max.ptw.gene	Filter option; minimum number of genes in a pathway; default is 5000 (pathway with >5000 genes are not included in the matrix)

Value

pathways	frequent pathways
tissues	frequent tissues

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
getHeatmap(result)
```

---

getMat *Extract tissue/geneset enrichment matrix from query result*

Description

This function extracts the enrichment matrix from eQTL list query result. The rows of the matrixs are pathways; and the columns of the matrixs are tissues/cell lines of the eQTL sets. P-Values from enrichment tests are summarized in this matrix

Usage

```
getMat(res, ...)

## S4 method for signature 'loci2pathResult'
getMat(res, test.method = c("gene", "eqtl",
  "glm"), filter.quantile = 0.5, max.ptw.gene = 5000)
```

Arguments

res	query result from function query.egset.list()
...	additional params
test.method	Choose which enrichment test should be used to retrive p-values from. Options include:"gene"(default, gene-based fisher's exact test),"eqtl" (eqtl based fisher's exact test), "glm" (ordered query)
filter.quantile	Filter option; choose the max quantile of all p-values being kept in the matrix; default is 0.5, which means p-values larger than median p-values are discarded
max.ptw.gene	Filter option; minimum number of genes in a pathway; default is 5000 (pathway with >5000 genes are not included in the matrix)

Value

p-value matrix collected from enrichment result table

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
mat <- getMat(result, test.method="gene")
```

---

getPathDescription *Extract description for enriched pathways from query result and geneSet object*

Description

This function extracts the pathway description from geneSet object.

Usage

```
getPathDescription(res, ...)

## S4 method for signature 'loci2pathResult'
getPathDescription(res, geneset)
```

Arguments

res	query result from function query.egset.list()
...	additional params
geneset	A geneSet object

Value

a vector of gene set description from geneSet description slot

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
path.des <- getPathDescription(result, biocarta)
```

`getPval` *Extract tissue/geneset enrichment p-value distribution from query result*

Description

This function extracts the enrichment p-value distribution from eQTL list query result. P-values from different tissues/cell types are organized, and QQ-plot is generated against uniform distribution

Usage

```
getPval(res, ...)

## S4 method for signature 'loci2pathResult'
getPval(res, test.method = c("gene", "eqtl",
  "glm"))
```

Arguments

`res`	query result from function query.egset.list()
`...`	additional params
`test.method`	Choose which enrichment test should be used to retrive p-values from. Options include:"gene"(default, gene-based fisher's exact test),"eqtl" (eqtl based fisher's exact test), "glm" (ordered query)

Value

generate pval distribution plot

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
getPval(result, test.method="gene")
```

getTissueDegree *Extract tissue degree from query result*

Description

This function extracts the tissue degree from eQTL list query result for each pathway.

Usage

```
getTissueDegree(res, ...)

## S4 method for signature 'loci2pathResult'
getTissueDegree(res, loci)
```

Arguments

res	query result from function query.egset.list()
...	additional params
loci	a list of eqtlSet; each member should be an eqtlSet object

Value

gene.tissue.map
: shows mapping:gene<->tissue

gene.tissue.degree
: shows tissue degree for each gene

mean.tissue.degree
: shows the average tissue digree for each pathway in the result table

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
tissue.degree=getTissueDegree(result, eset.list)
head(tissue.degree$gene.tissue.map)
head(tissue.degree$gene.tissue.degree)
head(tissue.degree$mean.tissue.degree)
```

getWordcloud *Plot word cloud using frequent terms of pathways and genes*

Description

This function draw the enrichment heatmap using wordcloud package.

Usage

```
getWordcloud(res, ...)

## S4 method for signature 'loci2pathResult'
getWordcloud(res, min.freq.tissue = 5,
  min.freq.gset = 5, max.words = 50)
```

Arguments

res	query result from function query.egset.list()
...	additional params
min.freq.tissue	minimum frequency of tissue/cell to be plotted in the word cloud
min.freq.gset	minimum frequency of geneset to be plotted in the word cloud
max.words	maximum words to be generated

Value

empty

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
getWordcloud(result, min.freq.tissue=2, min.freq.gset=1)
```

loci2pathResult-class *loci2pathResult Class*

Description

loci2pathResult Class contains information for the query result from query function query. Result object contains a ranked pathway table, and a vector of gene names that are associated with loci covered by query regions

Usage

```
resultTable(x)

coveredGene(x)

## S4 method for signature 'loci2pathResult'
resultTable(x)

## S4 method for signature 'loci2pathResult'
coveredGene(x)
```

Arguments

x	An geneSet object

Value

Object of CLass loci2pathResult

Slots

`resultTable` data.frame; contains enrichment statistics, summary of eQTL and gene numbers, pathway names and gene names, etc.

`coveredGene` list; each member is a vector of genes associated with one tissue, whose associating loci are covered by query regions

Examples

```
result <- query(query.gr=query.gr,
  loci=eset.list, path=biocarta)
result
resultTable(result) # a data.frame for enriched pathways
coveredGene(result)
```

---

query *Query enrichment in geneset through multiple eQTL sets.*

Description

This is the main function for loci2path query. Query can be made on either pathway enrichment or tissue-specificity, depending on the input Class. See **Details** for more.

Usage

```
query(query.gr, loci, path, ...)

## S4 method for signature 'GenomicRanges,list,ANY'
query(query.gr, loci, N = 2897310462)

## S4 method for signature 'GenomicRanges,eqtlSet,geneSet'
query(query.gr, loci, path,
  query.score = NULL, verbose = FALSE)

## S4 method for signature 'GenomicRanges,list,geneSet'
query(query.gr, loci, path,
  query.score = NULL, parallel = FALSE, verbose = FALSE)
```

Arguments

query.gr	a GenomicRange object, representing query regions
loci	a list of eqtlSet; each member should be an eqtlSet; Or it can be a single eqtlSet.
path	Pathways or geneSets to be tested for enrichment
...	additional params
N	the total number of non-N nucleotides in the genome; default N=2897310462 is for hg19
query.score	optional, set to NULL if the regions are not ordered. If the query regions are ordered, query.score is the quantity based on which the regions are ordered
verbose	bool; whether to show eqtlSet/geneSet summary information; default is FALSE
parallel	bool; whether to enable parallel computing; default is FALSE

Details

The user need to specify

1. Query region;
2. loci; one or more eQTL set; this is usually more than one eQTL set. Only multiple eQTL set derived from different cells/tissues will show cell/tissue specificity.
3. path; pre-defined Pathways, or gene sets. the gene sets that enrichment tests would be performed to.

`loci` must be provided; `path` is optional. When `path` is missing, the tissue-specificity query for the regions is performed.

The most common case for `loci` is an eQTL set list. This function perform enrichment test between one eQTL set and a group of gene sets. Usually query are based on eQTL set list, rather than only one eQTL set. Several result exploring functions (`getMat`, `getHeatmap`, `getPval`, etc...) are designed for query result from eQTL set list and gene sets. The class `loci2pathResult` is also designed for eQTL set list query result only. The result returns a `loci2pathResult` only the class of `loci` is a list of `eqtlSet`.

If user input one eQTL set as argument `loci`, a simple list object will be returned for specific research purpose.

Value

a data.frame showing the tissue enrichment of the query regions by binomial test.

a list; `result.table` is the major result table showing enrichment assessment; `cover.gene` is the vector showing the genes from the eqtl Sets covered by the query region(s)

a `loci2pathResult` class object

See Also

loci2pathResult

Examples

```
gr.tissue <- query(query.gr, eset.list)
#build one eqtlset
skin.eset <- eset.list$Skin
#query one egset
res.one <- query(query.gr, skin.eset, biocarta)
#enrichment result table
res.one$result.table
#all the genes associated with eQTLs covered by the query region
res.one$cover.gene
result <- query(query.gr=query.gr,
    loci=eset.list, path=biocarta)
#enrichment result table
resultTable(result)
#all the genes associated with eQTLs covered by the query region
coveredGene(result)
```

`query.gr` *eQTL geneset enrichment query demo data*

Description

Demo Data to show how to perform eQTL-geneset enrichment query.

Usage

```
data(loci2path.demo)
```

Format

An object of class `GRanges` of length 47.

Details

`eqtl.set.list` A list of eQTLset objects; eQTL data are collected from GTeX.

`biocarta` A Geneset object; Geneset are from Broad Institute's MSigDB: curated gene set, category 'cp': BIOCARTA

`query.gr` A Genomic Region object; Query regions are from immunoBase, crohn's disease.

Examples

```
data(loci2path.demo)
```

Index

∗ **cloud**
getWordcloud, 11
∗ **datasets**
biocarta, 2
eset.list, 4
query.gr, 14
∗ **heatmap**
getHeatmap, 6
∗ **result**
getMat, 7
getPval, 9
getTissueDegree, 10
∗ **word**
getWordcloud, 11

biocarta, 2

check.geneid, 3
coveredGene *(*loci2pathResult-class*)*, 11
coveredGene,loci2pathResult-method *(*loci2pathResult-class*)*, 11

description *(*geneSet-class*)*, 5
description,geneSet-method *(*geneSet-class*)*, 5

eqtlGene *(*eqtlSet-class*)*, 3
eqtlGene,eqtlSet-method *(*eqtlSet-class*)*, 3
eqtlId *(*eqtlSet-class*)*, 3
eqtlId,eqtlSet-method *(*eqtlSet-class*)*, 3
eqtlRange *(*eqtlSet-class*)*, 3
eqtlRange,eqtlSet-method *(*eqtlSet-class*)*, 3
eqtlSet *(*eqtlSet-class*)*, 3
eqtlSet-class, 3
eset.list, 4

geneSet *(*geneSet-class*)*, 5
geneSet-class, 5
geneSetList *(*geneSet-class*)*, 5
geneSetList,geneSet-method *(*geneSet-class*)*, 5
getHeatmap, 6
getHeatmap,loci2pathResult-method *(*getHeatmap*)*, 6
getMat, 7
getMat,loci2pathResult-method *(*getMat*)*, 7
getPathDescription, 8
getPathDescription,loci2pathResult-method *(*getPathDescription*)*, 8
getPval, 9
getPval,loci2pathResult-method *(*getPval*)*, 9
getTissueDegree, 10
getTissueDegree,loci2pathResult-method *(*getTissueDegree*)*, 10
getWordcloud, 11
getWordcloud,loci2pathResult-method *(*getWordcloud*)*, 11

loci2pathResult *(*loci2pathResult-class*)*, 11
loci2pathResult-class, 11

numGene *(*geneSet-class*)*, 5
numGene,geneSet-method *(*geneSet-class*)*, 5

query, 12
query,eqtlSet,geneSet-method *(*query*)*, 12
query,GenomicRanges,eqtlSet,geneSet-method *(*query*)*, 12
query,GenomicRanges,list,ANY-method *(*query*)*, 12
query,GenomicRanges,list,geneSet-method *(*query*)*, 12
query,list,geneSet-method *(*query*)*, 12
query,list-method *(*query*)*, 12
query.gr, 14

resultTable (loci2pathResult-class), 11
resultTable,loci2pathResult-method
(loci2pathResult-class), 11

tissue (eqtlSet-class), 3
tissue,eqtlSet-method (eqtlSet-class), 3