Package 'snpEnrichment'

October 1, 2015

Type Package

Title SNPs Enrichment Analysis

Version 1.7.0

Date 2015-10-01

Description Implements classes and methods for large scale SNP enrichment analysis (e.g. SNPs associated with genes expression in a GWAS signal).

License GPL (>= 2)

Depends R (>= 3.0.0), methods

Imports parallel, snpStats, grid, ggplot2, grDevices, graphics, stats, utils

URL https://github.com/mcanouil/snpEnrichment

Encoding UTF-8

BugReports https://github.com/mcanouil/snpEnrichment/issues

NeedsCompilation no

Author Mickael Canouil [aut, cre], Loic Yengo [ctb]

Maintainer Mickael Canouil <mickael.canouil@cnrs.fr>

Repository CRAN

Date/Publication 2015-10-01 23:18:57

R topics documented:

`snpEnrichment-package` *~ Overview: SNPs enrichment analysis ~*

Description

Implements classes and methods for large-scale SNP enrichment analysis (e.g. SNPs associated with genes expression in a GWAS signal).

Details

Package:	snpEnrichment
Title:	SNPs enrichment analysis
Author:	Mickael Canouil
Contributor:	Loic Yengo
Maintainer:	Mickael Canouil
License:	GPL (>= 2)
Depends:	R (>= 3.0.0), methods
Suggests:	grid, ggplot2
Imports:	parallel, snpStats
URL:	https://github.com/mcanouil/snpEnrichment
Encoding:	UTF-8

Note

Internal data management in 'snpEnrichment' use RefSNP (rs) IDs.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
###################
### 1. Prepare data
## Not run: snpInfoDir <- system.file("extdata/snpInfo",
                          package = "snpEnrichment")
signalFile <- system.file("extdata/Signal/toySignal.txt",
                          package = "snpEnrichment")

initFiles(pattern = "Chrom", snpInfoDir, signalFile, mc.cores = 1)

writeLD(pattern = "Chrom", snpInfoDir, signalFile,
        ldDir = NULL, ldThresh = 0.8, depth = 1000,
        mc.cores = 1)
## End(Not run)


################
### 2. Read data
## Not run: snpListDir <- system.file("extdata/List",
                          package = "snpEnrichment")
data(transcript)
transcriptFile <- transcript

toyData <- readEnrichment(pattern = "Chrom", signalFile,
                          transcriptFile, snpListDir,
                          snpInfoDir, distThresh = 1000,
                          sigThresh = 0.05, LD = TRUE,
                          ldDir = NULL, mc.cores = 1)
toyData
## End(Not run)


######################
### 3. Compute results
## Not run: reSample(object = toyData,
         nSample = 10,
         empiricPvalue = TRUE,
         MAFpool = c(0.05, 0.10, 0.2, 0.3, 0.4, 0.5),
         mc.cores = 1,
         onlyGenome = TRUE)
## End(Not run)


######################
```

```
### 4. Further analysis: Exclude SNP from original list.
## Not run: excludeFile <- c(
    "rs4376885", "rs17690928", "rs6460708", "rs13061537", "rs11769827",
    "rs12717054", "rs2907627", "rs1380109", "rs7024214", "rs7711972",
    "rs9658282", "rs11750720", "rs1793268", "rs774568", "rs6921786",
    "rs1699031", "rs6994771", "rs16926670", "rs465612", "rs3012084",
    "rs354850", "rs12803455", "rs13384873", "rs4364668", "rs8181047",
    "rs2179993", "rs12049335", "rs6079926", "rs2175144", "rs11564427",
    "rs7786389", "rs7005565", "rs17423335", "rs12474102", "rs191314",
    "rs10513168", "rs1711437", "rs1992620", "rs283115", "rs10754563",
    "rs10851727", "rs2173191", "rs7661353", "rs1342113", "rs7042073",
    "rs1567445", "rs10120375", "rs550060", "rs3761218", "rs4512977"
)
# OR
excludeFile <- system.file("extdata/Exclude/toyExclude.txt",
                           package = "snpEnrichment")

toyData_exclude <- excludeSNP(toyData, excludeFile, mc.cores = 1)

# Warning: compareEnrichment is in development!!
compareResults <- compareEnrichment(object.x = toyData,
                                    object.y = toyData_exclude,
                                    pattern = "Chrom",
                                    nSample = 10,
                                    empiricPvalue = TRUE,
                                    mc.cores = 1,
                                    onlyGenome = TRUE)
## End(Not run)


####################
### 5. Watch results
## Not run: show(toyData)
print(toyData)
head(getEnrichSNP(toyData, type = "xSNP"))

show(toyData_exclude)
print(toyData_exclude)
head(getEnrichSNP(toyData_exclude, type = "eSNP"))
## End(Not run)
```

Chromosome-class *Class* Chromosome

Description

This class is defined to summarize the enrichment analysis about a chromosome.

Objects from the Class

chromosome is defined to build an object of class Chromosome in order to compute an enrichment analysis. A Chromosome object contains the original data, a list of SNPs, some results and resampling data.

When created, an Chromosome object is "empty". readEnrichment initializes a Chromosome object with value from PLINK computation and user's files. In this step, only the fields "Data", "LD", "SNP" are filled. reSample fills the fields: Table, EnrichmentRatio, Z, PValue and Resampling of a Chromosome.

Note that if reSample is executed on an Chromosome every new resampling is added to the original ones, pre-existing statistics are erased and computed again with the new resampling set.

Slots

Data [data.frame]: a data.frame with 6 columns ("SNP", "PVALUE", "CHR", "MAF", "eSNP", "xSNP"). Where "eSNP" and "xSNP" are logical columns defining the lists of SNPs (extended or not).

LD [data.frame]: a data.frame which contains LD informations between SNPs (computed with writeLD or PLINK).

eSNP, xSNP [SNP]: contain a EnrichSNP object (whith slots: SNP, Table, EnrichmentRatio, Z, PValue and Resampling) for a list of SNPs (eSNP) and an extended one (xSNP).

Extends

Class EnrichSNP, directly.

Methods

chromosome(Data, LD, eSNP, xSNP): Generate and initialize a new Chromosome object.

object["slotName" :] Get the value of the field slotName.

object["slotName" <-value:] Set value to the field slotName.

show(object): Return the formatted values of Chromosome object.

Note

Chromosome object is not intended to be used alone on this version.\ It is a part of the Enrichment object.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
Data <- structure(
    list(
        SNP = c("rs4970420", "rs3766178",
                "rs10910030", "rs10910036",
                "rs2234167", "rs6661861"),
        PVALUE = c(0.9244, 0.167, 0.01177, 0.4267, 0.9728, 0.4063),
        CHR = c(1, 1, 1, 1, 1, 1),
        POS = c(1106473, 1478180, 2035684, 2183754, 2494330, 3043121),
        MAF = c(0.2149, 0.3117, 0.374, 0.3753, 0.1565, 0.06101),
        eSNP = c(0, 1, 1, 0, 0, 0),
        xSNP = c(0, 1, 1, 0, 0, 0)
    ),
    .Names = c("SNP", "PVALUE", "CHR", "POS", "MAF", "eSNP", "xSNP"),
    row.names = c("rs4970420", "rs3766178",
                  "rs10910030", "rs10910036",
                  "rs2234167", "rs6661861"),
class = "data.frame")

toyChr <- chromosome(Data = Data)
show(toyChr)
toyChr

toyChr <- chromosome()
toyChr["Data"] <- Data
toyChr

is.chromosome(toyChr) # TRUE
```

---

compareEnrichment *Compare enrichment analysis between two SNPs list*

Description

Compare the enrichment analysis between two set of SNPs. compareEnrichment compare two Enrichment objects.

Usage

```
compareEnrichment(object.x, object.y, pattern = "Chrom",
                  nSample = 100, empiricPvalue = TRUE,
                  mc.cores = 1, onlyGenome = TRUE)
```

Arguments

object.x, object.y	[Enrichment]: an Enrichment object fully filled (e.g. readEnrichment).
pattern	[character]: character string containing a expression to be matched with all chromosomes files (e.g."Chrom" for files which start by "Chrom" followed by the chromosome number).
nSample	[numeric]: the number of resampling done by reSample for p-values computation (minimum is 100).
empiricPvalue	[logical]: empiricPvalue=TRUE (default) compute PValue based on the null distribution (resampling). If empiricPvalue=TRUE, the empirical p-values are computed instead.
mc.cores	[numeric]: the number of cores to use (default is mc.cores=1), i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores.
onlyGenome	[logical]: onlyGenome=TRUE (default) compute resampling step for all chromosomes.

Value

Return a list of three elements:

object.xy	Enrichment object from the comparison between object.x and object.y.
object.x	Enrichment object passed in object.x with resampling data.
object.y	Enrichment object passed in object.y with resampling data.

Note

Still in development.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: data(toyEnrichment)

reSample(object = toyEnrichment,
         nSample = 10,
         empiricPvalue = TRUE,
```

```
          MAFpool = c(0.05, 0.10, 0.2, 0.3, 0.4, 0.5),
          mc.cores = 1,
          onlyGenome = TRUE)

excludeFile <- c(
    "rs7897180", "rs4725479", "rs315404", "rs17390391", "rs1650670",
    "rs6783390", "rs1642009", "rs4756586", "rs11995037", "rs4977345",
    "rs13136448", "rs4233536", "rs11151079", "rs2299657", "rs4833930",
    "rs1384", "rs7168184", "rs6909895", "rs7972667", "rs2293229",
    "rs918216", "rs6040608", "rs2817715", "rs13233541", "rs4486743",
    "rs2127806", "rs10912854", "rs1869052", "rs9853549", "rs448658",
    "rs2451583", "rs17483288", "rs10962314", "rs9612059", "rs1384182",
    "rs8049208", "rs12215176", "rs2980996", "rs1736976", "rs8089268",
    "rs10832329", "rs12446540", "rs7676237", "rs869922", "rs16823426",
    "rs1374393", "rs13268781", "rs11134505", "rs7325241", "rs7520109"
)
# OR
excludeFile <- system.file("extdata/Exclude/toyExclude.txt",
                           package = "snpEnrichment")

toyEnrichment_exclude <- excludeSNP(toyEnrichment, excludeFile, mc.cores = 1)

compareResults <- compareEnrichment(object.x = toyEnrichment,
                                    object.y = toyEnrichment_exclude,
                                    pattern = "Chrom",
                                    nSample = 10,
                                    empiricPvalue = FALSE,
                                    mc.cores = 1,
                                    onlyGenome = TRUE)
## End(Not run)
```

Enrichment-class *Class* Enrichment

Description

This class is defined to summarize the enrichment analysis on each chromosomes and the whole genome.

Objects from the Class

enrichment is defined to build an object of class Enrichment in order to compute an enrichment analysis. Enrichment is the object containing the results for all Chromosome object and for the whole genome.

When an Enrichment object is created, it contains a list of SNPs (e.g. eSNPs). All the others slots are "empty". After reSample is ran on an Enrichment object, the slots: Table, EnrichmentRatio, Z, PValue and Resampling are filled.

Note that if reSample is executed on an Enrichment every new resampling is added to the original ones, pre-existing statistics are erased and computed again with the new resampling set.

Slots

Loss [data.frame]: a four columns `data.frame`: "Rows", "Unique", "Intersect.Ref.Signal" and "CIS". This slot gives information on data losses.

Call [list]: each parameters used for the reading or resampling step are stored in this slot.

eSNP, xSNP [SNP]: contain a `EnrichSNP` object (whith slots: SNP, Table, EnrichmentRatio, Z, PValue and Resampling) for a list of SNPs (`eSNP`) and an extended one (`xSNP`).

Chromosomes [list(Chromosome)]: a list of 22 `Chromosome` objects.

Extends

Class `Chromosome`, directly.
Class `EnrichSNP`, directly.

Methods

enrichment(Loss, Call, eSNP, xSNP, Chromosomes): Generate and initialize a new `Enrichment` object.

object["slotName" :] Get the value of the field `slotName`.

object["slotName" <-value:] Set `value` to the field `slotName`.

show(object): Return the formatted values of `Enrichment` object.

print(object, what, type): Return a summary table of an `Enrichment` object.

reSample(object, nSample, MAFpool, mc.cores, onlyGenome): Compute P-Values based upon a resampling of SNPs (`eSNP` and `xSNP`) and update the `Enrichment` object.

excludeSNP(object, excludeFile, mc.cores): Excludes SNPs given in `excludeFile` from the original list of eSNPs (xSNPs). Then a new enrichment analysis is computed.

reset(object, "slotName"): Reset the field `slotName`.

plot(object, what, type, ggplot, pvalue): Plot p-values or Z-statistics convergence.

getEnrichSNP(object, type): Return eSNP/xSNP which are enriched as a `data.frame`.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : `snpEnrichment-package`
Classes : `Enrichment`, `Chromosome`, `EnrichSNP`
Methods : `plot`, `reSample`, `getEnrichSNP`, `excludeSNP`, `compareEnrichment`, `enrichment`, `is.enrichment`, `chromosome`, `is.chromosome`
Functions : `initFiles`, `writeLD`, `readEnrichment`

Examples

```
data(toyEnrichment)
toyEnrich <- enrichment()
show(toyEnrich)

toyEnrich["Loss"] <- toyEnrichment["Loss"]
toyEnrich["Loss"]

toyEnrich <- enrichment(Loss = toyEnrichment["Loss"],
                        eSNP = toyEnrichment["eSNP"])
toyEnrich <- enrichment(Loss = toyEnrichment["Loss"])

## Not run: reSample(object = toyEnrichment,
         nSample = 10,
         empiricPvalue = TRUE,
         MAFpool = c(0.05, 0.10, 0.2, 0.3, 0.4, 0.5),
         mc.cores = 1,
         onlyGenome = TRUE)
print(toyEnrichment)

excludeFile <- c(
"rs7897180", "rs4725479", "rs315404", "rs17390391", "rs1650670",
    "rs6783390", "rs1642009", "rs4756586", "rs11995037", "rs4977345",
    "rs13136448", "rs4233536", "rs11151079", "rs2299657", "rs4833930",
    "rs1384", "rs7168184", "rs6909895", "rs7972667", "rs2293229",
    "rs918216", "rs6040608", "rs2817715", "rs13233541", "rs4486743",
    "rs2127806", "rs10912854", "rs1869052", "rs9853549", "rs448658",
    "rs2451583", "rs17483288", "rs10962314", "rs9612059", "rs1384182",
    "rs8049208", "rs12215176", "rs2980996", "rs1736976", "rs8089268",
    "rs10832329", "rs12446540", "rs7676237", "rs869922", "rs16823426",
    "rs1374393", "rs13268781", "rs11134505", "rs7325241", "rs7520109"
)

toyEnrichment_exclude <- excludeSNP(toyEnrichment, excludeFile, mc.cores = 1)
print(toyEnrichment_exclude)
## End(Not run)
```

EnrichSNP-class *Class* "EnrichSNP"

Description

This class is defined to summarize the enrichment analysis. It's a part of Chromosome and Enrichment classes.

Slots

List [vector(character)]: a list of SNPs used to compute enrichment (e.g. eSNP or xSNP).

Table [matrix]: Contingency table with SNPs (columns) and P-Values from signal (rows).

EnrichmentRatio [numeric]: Enrichment Ratio is computed on the contingency table (`Table` slot).

Z [numeric]: A statistic computed from `EnrichmentRatio` and resampling results.

PValue [numeric]: P-Value associated with the statistic `Z`.

Resampling [matrix]: A matrix with by row, the contingency table and the odds ratio for each resampling.

Methods

object["slotName"] :] Get the value of the field `slotName`.

object["slotName"] <-value:] Set `value` to the field `slotName`.

show(object): Return the formatted values of `EnrichSNP` object.

Note

`EnrichSNP` object is not intended to be use directly by user. It is a part of the `Enrichment` and `Chromosome` object.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : `snpEnrichment-package`
Classes : `Enrichment`, `Chromosome`, `EnrichSNP`
Methods : `plot`, `reSample`, `getEnrichSNP`, `excludeSNP`, `compareEnrichment`, `enrichment`, `is.enrichment`, `chromosome`, `is.chromosome`
Functions : `initFiles`, `writeLD`, `readEnrichment`

---

excludeSNP *Exclude SNPs from Enrichment analysis*

Description

Remove a specify set of SNPs and compute a new enrichment analysis.

Usage

```
excludeSNP(object, excludeFile, mc.cores = 1)
```

Arguments

`object`	[Enrichment]: an `Enrichment` object filled by `reSample`.
`excludeFile`	[vector(character)]: a list of SNPs to remove from a previous enrichment analysis. A path to a file which the first column are the SNPs.
`mc.cores`	[numeric]: the number of cores to use (default is `mc.cores=1`), i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores.

Value

Return the object given in argument where lists of SNPs are updated by removing SNPs in `excludeFile`.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: data(toyEnrichment)
excludeFile <- c(
    "rs4376885", "rs17690928", "rs6460708", "rs13061537", "rs11769827",
    "rs12717054", "rs2907627", "rs1380109", "rs7024214", "rs7711972",
    "rs9658282", "rs11750720", "rs1793268", "rs774568", "rs6921786",
    "rs1699031", "rs6994771", "rs16926670", "rs465612", "rs3012084",
    "rs354850", "rs12803455", "rs13384873", "rs4364668", "rs8181047",
    "rs2179993", "rs12049335", "rs6079926", "rs2175144", "rs11564427",
    "rs7786389", "rs7005565", "rs17423335", "rs12474102", "rs191314",
    "rs10513168", "rs1711437", "rs1992620", "rs283115", "rs10754563",
    "rs10851727", "rs2173191", "rs7661353", "rs1342113", "rs7042073",
    "rs1567445", "rs10120375", "rs550060", "rs3761218", "rs4512977"
)
# OR
excludeFile <- system.file("extdata/Exclude/toyExclude.txt",
                           package = "snpEnrichment")

toyEnrichment_exclude <- excludeSNP(toyEnrichment, excludeFile, mc.cores = 1)
toyEnrichment_exclude
## End(Not run)
```

GC *Full Garbage Collection*

Description

GC performs garbage collection until free memory indicators show no change.

Usage

```
GC(verbose = getOption("verbose"), reset=FALSE)
```

Arguments

verbose	[logical]: if TRUE, the garbage collection prints statistics about cons cells and the space allocated for vectors.
reset	[logical]: if TRUE the values for maximum space used are reset to the current values.

Value

GC returns a matrix with rows "Ncells" (_cons cells_), usually 28 bytes each on 32-bit systems and 56 bytes on 64-bit systems, and "Vcells" (_vector cells_, 8 bytes each), and columns "used" and "gc trigger", each also interpreted in megabytes (rounded up to the next 0.1Mb).

If maxima have been set for either "Ncells" or "Vcells", a fifth column is printed giving the current limits in Mb (with NA denoting no limit).

The final two columns show the maximum space used since the last call to GC(reset=TRUE) (or since R started).

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

The R Internals manual.

Memory on R's memory management, and gctorture if you are an R developer.

reg.finalizer for actions to happen at garbage collection.

Examples

```
GC() # - do it now
x <- integer(100000); for(i in 1:18) x <- c(x,i)
GC(TRUE)
GC(reset=TRUE)
```

---

getEnrichSNP *Get all eSNP/xSNP which are enriched*

Description

getEnrichSNP get all eSNP/xSNP in a Enrichment object which are significant in the signal according to sigThresh defined in readEnrichment.

Usage

```
## S4 method for signature 'Enrichment'
getEnrichSNP(object, type = "eSNP")
```

Arguments

object	[Enrichment]: an object of class Enrichment.
type	[character]: extract eSNP or xSNP data.

Value

Return a data.frame with eSNP/xSNP which are enriched in signal given to signalFile in function readEnrichment.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: data(toyEnrichment)
eSNPenriched <- getEnrichSNP(object = toyEnrichment, type = "eSNP")
head(eSNPenriched)
## End(Not run)
```

---

initFiles *Initialize files for enrichment analysis*

Description

initFiles create several files needed to run readEnrichment. ".frq" and ".signal" are created with PLINK. LD computation can be run with writeLD or with PLINK.

Usage

```
initFiles(pattern = "Chrom", snpInfoDir, signalFile, mc.cores = 1)
```

Arguments

pattern	[character]: character string containing a expression to be matched with all chromosomes files (e.g."Chrom" for files which start by "Chrom" followed by the chromosome number).
snpInfoDir	[character]: character string naming a directory containing the reference data in a PLINK format (*.bed, *.bim and *.fam).

signalFile [character]: the name of the signal file which the data are to be read from (2 columns: "SNP" and "PVALUE"). Each row of the table appears as one line of the file. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, getwd. The fields separator character have to be a space " " or a tabulation "\t".

mc.cores [numeric]: the number of cores to use (default is mc.cores=1), i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores.

Value

This function writes several files, in the temporary directory (defined in R_SESSION_TMPDIR), nothing else is returned. These files are used to build an Enrichment object by readEnrichment in order to compute enrichment analysis (reSample).

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: snpInfoDir <- system.file("extdata/snpInfo",
                          package = "snpEnrichment")
signalFile <- system.file("extdata/Signal/toySignal.txt",
                          package = "snpEnrichment")
initFiles(pattern = "Chrom",
          snpInfoDir,
          signalFile,
          mc.cores = 1)
## End(Not run)
```

---

is.chromosome *Is an Chromosome object*

Description

'is.chromosome' returns 'TRUE' if 'x' is an Chromosome object and 'FALSE' otherwise.

Usage

```
is.chromosome(object)
```

Arguments

object	[ANY]: object to be tested.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
a <- chromosome()
c <- chromosome()
is.chromosome(list())               # FALSE
is.chromosome(1)                    # FALSE
is.chromosome(a)                    # TRUE
is.chromosome(c(a, c))              # TRUE TRUE
is.chromosome(list(a, b = "char"))  # TRUE FALSE
is.chromosome(c(a, b = list(12, c))) # TRUE FALSE TRUE
```

---

is.enrichment *Is an Enrichment object*

Description

'is.enrichment' returns 'TRUE' if 'x' is an Enrichment object and 'FALSE' otherwise.

Usage

```
is.enrichment(object)
```

Arguments

object	[ANY]: object to be tested.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
a <- enrichment()
c <- enrichment()
is.enrichment(list())               # FALSE
is.enrichment(1)                    # FALSE
is.enrichment(a)                    # TRUE
is.enrichment(c(a, c))              # TRUE TRUE
is.enrichment(list(a, b = "char"))  # TRUE FALSE
is.enrichment(c(a, b = list(12, c))) # TRUE FALSE TRUE
```

---

mclapply2 *Parallel Versions of* lapply *with cores and memory control*

Description

mclapply2 is a mclapply modification from parallel package, to avoid a memory overload. The maximum number of cores is computed depending on the amount of memory used by the parent process and the amount of free memory available on the machine. Note: This number is underestimated.

Usage

```
mclapply2(X, FUN, ..., mc.preschedule = TRUE, mc.set.seed = TRUE,
          mc.silent = FALSE, mc.cores = getOption("mc.cores", 2L),
          mc.cleanup = TRUE, mc.allow.recursive = TRUE)
```

Arguments

X	a vector (atomic or list) or an expressions vector. Other objects (including classed objects) will be coerced by as.list.
FUN	the function to be applied to (mclapply) each element of X or (mcmapply) in parallel to ....
...	For mclapply, optional arguments to FUN.
mc.preschedule	if set to TRUE then the computation is first divided to (at most) as many jobs are there are cores and then the jobs are started, each job possibly covering more than one value. If set to FALSE then one job is forked for each value of X. The former is better for short computations or large number of values in X, the latter is better for jobs that have high variance of completion time and not too many values of X compared to mc.cores.

mc.set.seed — See mcparallel.

mc.silent — if set to TRUE then all output on 'stdout' will be suppressed for all parallel processes forked ('stderr' is not affected).

mc.cores — The number of cores to use, i.e. at most how many child processes will be run simultaneously. The option is initialized from environment variable MC_CORES if set. Must be at least one, and parallelization requires at least two cores.

mc.cleanup — if set to TRUE then all children that have been forked by this function will be killed (by sending SIGTERM) before this function returns. Under normal circumstances mclapply waits for the children to deliver results, so this option usually has only effect when mclapply is interrupted. If set to FALSE then child processes are collected, but not forcefully terminated. As a special case this argument can be set to the number of the signal that should be used to kill the children instead of SIGTERM.

mc.allow.recursive — Unless true, calling mclapply in a child process will use the child and not fork again.

Value

A list of the same length as X and named by X.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

mclapply from package parallel.

---

plot-methods *Plot method (S4) for* Enrichment *object*

Description

plot is a generic function for plotting of R objects. The function invokes particular methods which depend on the class of the first argument.

Usage

```
## S4 method for signature 'Enrichment'
plot(x, what = "Genome", type = c("eSNP", "xSNP"),
           ggplot = FALSE, pvalue = TRUE, ...)
```

Arguments

x	[Enrichment]: an object of class Enrichment which the Z statistics or p-values have to be drawn.
what	[character or vector(numeric)]: default what="Genome") plot Z statistics or p-values for genome only (what must be: "All", "Genome" or numeric vector).
type	[vector(character)]: plot the selected analysis for "eSNP" and/or "xSNP".
ggplot	[logical]: use ggplot (default ggplot=FALSE) instead of classic plot method.
pvalue	[logical]: if TRUE, p-value convergense is plotted. Otherwise, Z statistic is plotted.
...	[any]: Arguments to be passed to methods, such as graphical parameters (see par)

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: data(toyEnrichment)
reSample(toyEnrichment, 10)
plot(toyEnrichment)
## End(Not run)
```

---

print-methods *Print method (S4)*

Description

print is a generic function used to print results.

Usage

```
## S4 method for signature 'Enrichment'
print(x, what = "Genome", type = c("eSNP", "xSNP"))

## S4 method for signature 'Chromosome'
print(x, type = c("eSNP", "xSNP"))
```

Arguments

x	[Enrichment or Chromosome]: an object of class Enrichment or Chromosome.
what	[character or numeric]: what="Genome" (default) to print results as a matrix. what could be "All", "Genome" or a numeric from 1 to 22 (numeric vector is allowed).
type	[character]: select if results for "eSNP" and/or "xSNP" should be print.

Value

Return a matrix for classes Enrichment and Chromosome.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
data(toyEnrichment)
print(toyEnrichment, "All", type = "eSNP")
print(toyEnrichment, "Genome")
print(toyEnrichment, 1)
```

---

readEnrichment *Read and create EnrichmentRatio object*

Description

Read files created by initFiles and create an Enrichment object.

Usage

```
readEnrichment(pattern = "Chrom", signalFile,
               transcriptFile = "FALSE", snpListDir,
               snpInfoDir, distThresh = 1000,
               sigThresh = 0.05, LD = FALSE, ldDir = NULL,
               mc.cores = 1)
```

Arguments

pattern	[character]: character string containing a expression to be matched with all chromosomes files (e.g."Chrom" for files which start by "Chrom" followed by the chromosome number).
signalFile	[character]: the name of the signal file which the data are to be read from (2 columns: "SNP" and "PVALUE"). Each row of the table appears as one line of the file. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, getwd. The fields separator character have to be a space " " or a tabulation "\t".
transcriptFile	[character or data.frame]: character string naming a file or a data.frame with four columns: Chromomosome, trancript's name, Starting and Ending positions. data(trancript) can be use as parameters. Default is FALSE.
snpListDir	[character]: character string naming a directory containing a list of SNPs for one or several chromosomes. snpListDir can be a single file with at least two columns: chromosome and rs name.
snpInfoDir	[character]: character string naming a directory containing the reference data in a PLINK format (*.bed, *.bim and *.fam).
distThresh	[numeric]: maximal distance (kb) between SNP and gene. distThresh is used if transcriptFile is set.
sigThresh	[numeric]: statistical threshold for signal (e.g. sigThresh = 0.05 for a given GWAS signal) used to compute an Enrichment Ratio.
LD	[logical]: LD=TRUE (default is FALSE) read LD compute with writeLD function or with PLINK. Note that, this setting can increase the computation's time, depending on number of SNPs in the signal file.
ldDir	[character]: character string naming a directory where the linkage disequilibrium files should be read (default ldDir=NULL is in temporary directory). LD files can be the LD output from plink.
mc.cores	[numeric]: the number of cores to use (default is mc.cores=1), i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores.

Value

Return an object of class Enrichment partly filled.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: snpListDir <- system.file("extdata/List",
                          package = "snpEnrichment")
signalFile <- system.file("extdata/Signal/toySignal.txt",
                          package = "snpEnrichment")
snpInfoDir <- system.file("extdata/snpInfo", package = "snpEnrichment")
data(transcript)
transcriptFile <- transcript

initFiles(pattern = "Chrom", snpInfoDir, signalFile, mc.cores = 1)
toyData <- readEnrichment(pattern = "Chrom",
                          signalFile,
                          transcriptFile,
                          snpListDir,
                          snpInfoDir,
                          distThresh = 1000,
                          sigThresh = 0.05,
                          LD = FALSE,
                          ldDir = NULL,
                          mc.cores = 1)
toyData
## End(Not run)
```

---

reSample *Compute enrichment analysis on an* Enrichment *object*

Description

After initFiles and readEnrichment has been run. reSample computes a statistic value and a p-value for each chromosomes and for the whole genome.

Usage

```
## S4 method for signature 'Enrichment'
reSample(object, nSample = 100,
         empiricPvalue = TRUE,
         MAFpool = c(0.05, 0.10, 0.2, 0.3, 0.4, 0.5),
         mc.cores = 1, onlyGenome = TRUE)

## S4 method for signature 'Chromosome'
reSample(object, nSample = 100,
         empiricPvalue = TRUE, sigThresh = 0.05,
         MAFpool = c(0.05, 0.10, 0.2, 0.3, 0.4, 0.5),
         mc.cores = 1)
```

Arguments

object	[Enrichment or Chromosome]: an object to be updated. It is intended, an object returned by the readEnrichment function.
nSample	[numeric]: the number of resampling done by reSample for p-values computation (minimum is 100).
empiricPvalue	[logical]: empiricPvalue=TRUE (default) compute PValue based on the null distribution (resampling). If empiricPvalue=TRUE, the empirical p-values are computed instead.
sigThresh	[numeric]: statistical threshold for signal (e.g. sigThresh = 0.05 for a given GWAS signal) used to compute an Enrichment Ratio.
MAFpool	[vector(numeric)]: either a numeric vector giving the breaks points of intervals into which SNP's MAF (Minor Allele Frequency) is to be split.
mc.cores	[numeric]: the number of cores to use (default is mc.cores=1), i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores.
onlyGenome	[logical]: onlyGenome=TRUE (default) compute resampling step for all chromosomes.

Value

Return the object given in argument, updated by the resampling results.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: data(toyEnrichment)
reSample(object = toyEnrichment,
         nSample = 10,
         empiricPvalue = TRUE,
         MAFpool = c(0.05, 0.10, 0.2, 0.3, 0.4, 0.5),
         onlyGenome = TRUE)
toyEnrichment
## End(Not run)
```

toyEnrichment-dataset *Toy dataset with SNP data*

Description

This data set gives an Enrichment object after, initFiles and readEnrichment is ran. Compute LD for all SNPs in snpListDir files two by two. Genome Build 37.3 (hg19).

Usage

```
data(toyEnrichment)
toyEnrichment
```

Format

See class Enrichment for details about the format.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

transcript-dataset *Transcript information in order to check the CIS status for SNPs*

Description

This dataset is used by readEnrichment and compareEnrichment in order to check the CIS status for each SNP of signal. Genome Build 37.3 (hg19).

Usage

```
data(transcript)
transcript
```

Format

See class readEnrichment and compareEnrichment for details about how to use this dataset.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

---

writeLD *Linkage Disequilibrium (LD) computation with PLINK*

Description

writeLD write a '.ld' file for each chromosomes which contains the LD (r^2).

Usage

```
writeLD(pattern = "Chrom", snpInfoDir, signalFile,
        ldDir = NULL, ldThresh = 0.8, depth = 1000,
        mc.cores = 1)
```

Arguments

pattern	[character]: character string containing a expression to be matched with all chromosomes files (e.g."Chrom" for files which start by "Chrom" followed by the chromosome number).
snpInfoDir	[character]: character string naming a directory containing the reference data in a PLINK format (*.bed, *.bim and *.fam).
signalFile	[character]: the name of the signal file which the data are to be read from (2 columns: "SNP" and "PVALUE"). Each row of the table appears as one line of the file. If it does not contain an _absolute_ path, the file name is _relative_ to the current working directory, getwd. The fields separator character have to be a space " " or a tabulation "\t".
ldDir	[character]: character string naming a directory where the linkage Disequilibrium files should be written (default ldDir=NULL is in temporary directory).
ldThresh	[numeric]: threshold value for LD calculation.
depth	[numeric]: this parameter is mandatory and controls the maximum lag between SNPs considered.
mc.cores	[numeric]: the number of cores to use (default is mc.cores=1), i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores.

Value

One ".ld" file per chromosome is returned by writeLD in snpInfoDir directory.

Note

The LD computation can take a long time depending on number of SNPs in signalFile. It is recommended to save LD results in a directory (ldDir) which is not a temporary directory.

Author(s)

Mickael Canouil <mickael.canouil@good.ibl.fr>

See Also

Overview : snpEnrichment-package
Classes : Enrichment, Chromosome, EnrichSNP
Methods : plot, reSample, getEnrichSNP, excludeSNP, compareEnrichment, enrichment, is.enrichment, chromosome, is.chromosome
Functions : initFiles, writeLD, readEnrichment

Examples

```
## Not run: signalFile <- system.file("extdata/Signal/toySignal.txt",
                          package = "snpEnrichment")
snpInfoDir <- system.file("extdata/snpInfo",
                          package = "snpEnrichment")
writeLD(pattern = "Chrom", snpInfoDir, signalFile,
        ldDir = NULL, ldThresh = 0.8, mc.cores = 1)
## End(Not run)
```

Index