Package: diggit (via r-universe)

July 2, 2024

Version 1.37.0

Date 2014-08-22

Title Inference of Genetic Variants Driving Cellular Phenotypes

Author Mariano J Alvarez <reef103@gmail.com>

Maintainer Mariano J Alvarez <reef103@gmail.com>

Depends R (>= 3.0.2), Biobase, methods

Imports ks, viper(>= 1.3.1), parallel

Suggests diggitdata

Description Inference of Genetic Variants Driving Cellullar Phenotypes by the DIGGIT algorithm

License file LICENSE

biocViews SystemsBiology, NetworkEnrichment, GeneExpression, FunctionalPrediction, GeneRegulation

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

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

RemoteRef HEAD

RemoteSha a40b8a99be8a9601c81d6989991edb36ecf93f16

Contents

aecdf *Approximate empirical commulative distribution function*

Description

This function generates an empirical null model that computes a normalized statistics and p-value

Usage

```
aecdf(dnull, symmetric = FALSE)
```

Arguments

`dnull`	Numerical vector representing the null model
`symmetric`	Logical, whether the distribution should be treated as symmetric around zero and only one tail should be approximated

Value

function with two parameters, `x` and `alternative`

aqtl *Inference of aQTL*

Description

This function infers aQTLs from F-CNVs and VIPER activity

Usage

```
aqtl(x, ...)

## S4 method for signature 'diggit'
aqtl(x, mr = 0.01, mr.adjust = c("none", "fdr",
  "bonferroni"), fcnv = 0.01, fcnv.adjust = c("none", "fdr", "bonferroni"),
  method = c("spearman", "mi", "pearson", "kendall"), mindy = FALSE,
  cores = 1, verbose = TRUE)
```

Arguments

`x`	Object of class diggit
`...`	Additional parameters to pass to the function
`mr`	Either a numerical value between 0 and 1 indicating the p-value threshold for the Master Regulator (MR) selection, or a vector of character strings listing the MRs

`mr.adjust` Character string indicating the multiple hypothesis test correction for the MRs

`fcnv` Either a numerical value between 0 and 1 indicating the p-value threshold for the F-CNV, or a vector of character strings listing the F-CNVs

`fcnv.adjust` Character string indicating the multiple hypothesis test correction for the F-CNVs

`method` Character string indicating the method for computing the association between F-CNV and regulator activity (aQTL analysis)

`mindy` Logical, whether only post-translational modulators of each evaluated TF should be considered as putative genetic driver

`cores` Integer indicating the number of cores to use (1 for Windows-based systems)

`verbose` Logical, whether progress should be reported

Value

Updated diggit object with viper and aqtl slots

Examples

```
data(gbm.expression, package="diggitdata")
data(gbm.cnv, package="diggitdata")
data(gbm.aracne, package="diggitdata")
dobj <- diggitClass(expset=gbmExprs, cnv=gbmCNV, regulon=gbmTFregulon)
dobj <- fCNV(dobj)
dobj <- aqtl(dobj, mr=c("CEBPD", "STAT3"), fcnv.adjust="fdr")
dobj
diggitAqtl(dobj)[, 1:4]
```

`conditional` *Conditional analysis of CNVs*

Description

This function performs the conditional analysis of fCNVs

Usage

```
conditional(x, ...)

## S4 method for signature 'diggit'
conditional(x, pheno = "cond", group1, group2 = NULL,
  cnv = 0.2, mr = 0.01, mr.adjust = c("none", "fdr", "bonferroni"),
  modul = 0.01, modul.adjust = c("none", "fdr", "bonferroni"),
  fet.pval = 0.05, cores = 1, verbose = TRUE)
```

Arguments

x Object of class diggit

... Additional parameters to pass to the function

pheno Character string indicating the feature for sample groups

group1 Character string indicating the treatment group

group2 Optional character string indicating the reference group

cnv Single number or vector of two numbers indicating the thresholds for CNVs

mr Either vector of character strings indicating the MR genes, or number indicating the corrected p-value threshold for selecting the MRs

mr.adjust Character string indicating the multiple-hypothesis correction to apply to the MR p-values

modul Number indicating the p-value threshold for a modulator to be considered associated with the MR activity

modul.adjust Character string indicating the multiple-hypothesis correction to apply to the aQTL results

fet.pval Number indicating the FET p-value threshold for the association between CNVs and sample groups

cores Integer indicating the number of cores to use (1 for Windows-based systems)

verbose Logical, whether progress should be reported

Value

Object of class diggit with conditional analysis results

Examples

```
data(gbm.expression, package="diggitdata")
data(gbm.cnv, package="diggitdata")
data(gbm.aracne, package="diggitdata")
dobj <- diggitClass(expset=gbmExprs, cnv=gbmCNV, regulon=gbmTFregulon)
dobj <- fCNV(dobj)
dobj <- aqtl(dobj, mr=c("CEBPD", "STAT3"), fcnv.adjust="fdr", verbose=FALSE)
dobj <- conditional(dobj, pheno="subtype", group1="MES", group2="PN", mr="STAT3", verbose=FALSE)
dobj
```

correlation *Correlation test*

Description

This function computes the correlation between x and y given both are numeric vectors, between the columns of x if it is a numeric matrix, or between the columns of x and y if both are numeric matrixes

Usage

```
correlation(x, y = NULL, method = c("pearson", "spearman", "kendall"),
  pairwise = FALSE)
```

Arguments

`x`	Numeric vector or matrix
`y`	Optional numeric vector or matrix
`method`	Character string indicating the correlation method
`pairwise`	Logical, wether columns of x and y should be compared in a pairwise manner. x and y must have the same number of columns

Details

This function computes correlation and associated p-values

Value

Numeric value, vector or matrix of results

Examples

```
x <- seq(0, 10, length=50)
y <- x+rnorm(length(x), sd=2)
correlation(x, y)
```

`diggit-class` *The diggit class*

Description

This class stores parameters and results of the diggit algorithm

This function generates diggit class objects

Usage

```
diggitClass(expset = NULL, cnv = NULL, regulon = NULL, mindy = NULL,
  fcnv = NULL, mr = NULL, viper = NULL, aqtl = NULL,
  conditional = NULL)
```

Arguments

`expset` ExpressionSet object or numeric matrix of expression data, with features in rows and samples in columns

`cnv` Numeric matrix of CNV data

`regulon` Regulon class object containing the transcriptional interactome

`mindy` Regulon class object containing the post-translational interactome

`fcnv` Vector of F-CNV p-values

`mr` Vector of master regulator Z-score (NES)

`viper` Numeric matrix of VIPER results

`aqtl` Numeric matrix of aQTL p-values

`conditional` List containing the conditional analysis results

Details

see `diggit-methods` `for related methods`

Value

Object of class diggit

Slots

`expset`**:** ExpressionSet object containing the gene expression data

`cnv`**:** Matrrix containing the CNV data

`regulon`**:** Regulon object containing the transcriptional interactome

`mindy`**:** Regulon object containing the post-translational interactome

`fcnv`**:** Numeric vector containing the p-values for functional CNVs

`mr`**:** Numeric vector of normalized enrichment scores for the MARINa analysis

`viper`**:** Numeric matrix of normalized enrichment scores for the VIPER analysis

`aqtl`**:** Numeric matrix of association p-values for the aQTL analysis

`conditional`**:** List containing the conditional analysis results

Examples

```
data(gbm.expression, package="diggitdata")
data(gbm.aracne, package="diggitdata")
dobj <- diggitClass(expset=gbmExprs, regulon=gbmTFregulon)
print(dobj)
```

fCNV *Inference of functional CNVs*

Description

This function infers functional CNVs by computing their association with gene expression

Usage

```
fCNV(x, ...)

## S4 method for signature 'diggit'
fCNV(x, expset = NULL, cnv = NULL,
  method = c("spearman", "mi", "pearson", "kendall"), cores = 1,
  verbose = TRUE)

## S4 method for signature 'ExpressionSet'
fCNV(x, cnv, method = c("spearman", "mi", "pearson",
  "kendall"), cores = 1, verbose = TRUE)

## S4 method for signature 'matrix'
fCNV(x, cnv, method = c("spearman", "mi", "pearson",
  "kendall"), cores = 1, verbose = TRUE)

## S4 method for signature 'data.frame'
fCNV(x, cnv, method = c("spearman", "mi", "pearson",
  "kendall"), cores = 1, verbose = TRUE)
```

Arguments

`x`	Object of class diggit, expressionSet object or numeric matrix of expression data, with features in rows and samples in columns
`...`	Additional arguments
`expset`	Optional numeric matrix of expression data
`cnv`	Optional numeric matrix of CNVs
`method`	Character string indicating the method for computing the association between CNVs and expression
`cores`	Integer indicating the number of cores to use (1 for Windows-based systems)
`verbose`	Logical, whether to report analysis progress

Value

Objet of class diggit with updated fCNV slot

Examples

```
data(gbm.expression, package="diggitdata")
data(gbm.cnv, package="diggitdata")
genes <- intersect(rownames(gbmExprs), rownames(gbmCNV))[1:100]
gbmCNV <- gbmCNV[match(genes, rownames(gbmCNV)), ]
dgo <- diggitClass(expset=gbmExprs, cnv=gbmCNV)

dgo <- fCNV(dgo)
dgo
diggitFcnv(dgo)[1:5]
dgo <- fCNV(gbmExprs, gbmCNV)
print(dgo)
diggitFcnv(dgo)[1:5]
dgo <- fCNV(exprs(gbmExprs), gbmCNV)
dgo
diggitFcnv(dgo)[1:5]
dgo <- fCNV(as.data.frame(exprs(gbmExprs)), gbmCNV)
dgo
diggitFcnv(dgo)[1:5]
```

marina *Inference of Master Regulators*

Description

This function infers the master regulators for the transition between two phenotypes

Usage

```
marina(x, ...)

## S4 method for signature 'matrix'
marina(x, y = NULL, mu = 0, regulon, per = 1000,
  cores = 1, verbose = TRUE)

## S4 method for signature 'ExpressionSet'
marina(x, pheno = "cond", group1, group2 = NULL,
  mu = 0, regulon, per = 1000, cores = 1, verbose = TRUE)

## S4 method for signature 'diggit'
marina(x, pheno, group1, group2 = NULL, mu = 0,
  regulon = NULL, per = 1000, cores = 1, verbose = TRUE)
```

Arguments

`x`	Object of class diggit, expressionSet object or numerical matrix containing the test samples
`...`	Additional arguments

`y`	Numerical matrix containing the control samples
`mu`	Number indicating the control mean when `y` is ommited
`regulon`	Transcriptional interactome
`per`	Interger indicating the number of permutations to compute the marina null model
`cores`	Integer indicating the number of cores to use (1 for Windows-based systems)
`verbose`	Logical, whether progress should be reported
`pheno`	Character string indicating the phenotype data to use
`group1`	Vector of character strings indicating the category from phenotype `pheno` to use as test group
`group2`	Vector of character strings indicating the category from phenotype `pheno` to use as control group

Value

Updated diggit object with Master Regulator results

Examples

```
cores <- 3*(Sys.info()[1] != "Windows")+1
data(gbm.expression, package="diggitdata")
data(gbm.aracne, package="diggitdata")

eset <- exprs(gbmExprs)
samples <- pData(gbmExprs)[["subtype"]]
x <- eset[, samples=="MES"]
y <- eset[, samples=="PN"]
dgo <- marina(x, y, regulon=gbmTFregulon, per=100, cores=cores)
dgo
diggitMR(dgo)[1:5]
dgo <- marina(gbmExprs, pheno="subtype", group1="MES", group2="PN", regulon=gbmTFregulon, per=100, cores=cores)
dgo
diggitMR(dgo)[1:5]
x <- diggitClass(expset=gbmExprs, regulon=gbmTFregulon)
dgo <- marina(x, pheno="subtype", group1="MES", group2="PN", per=100, cores=cores)
dgo
diggitMR(dgo)[1:5]
```

`mutualInfo` *Mutual information*

Description

This function estimates the mutual information between x and y given both are numeric vectors, between the columns of x if it is a numeric matrix, or between the columns of x and y if both are numeric matrixes

Usage

```
mutualInfo(x, y = NULL, per = 0, pairwise = FALSE, bw = 100,
  cores = 1, verbose = TRUE)
```

Arguments

`x`	Numeric vector or matrix
`y`	Optional numeric vector or matrix
`per`	Integer indicating the number of permutations to compute p-values
`pairwise`	Logical, wether columns of x and y should be compared in a pairwise maner. x and y must have the same number of columns
`bw`	Integer indicating the grid size for integrating the joint probability density
`cores`	Integer indicating the number of cores to use (1 for Windows-based systems)
`verbose`	Logical, whether progression bars should be shown

Details

This function estimates the mutual information between continuous variables using a fix bandwidth implementation

Value

Numeric value, vector or matrix of results

Examples

```
x <- seq(0, pi, length=100)
y <- 5*sin(x)+rnorm(100)
cor.test(x, y)
mutualInfo(x, y, per=100)
```

`plot,diggit-method` *Diggit plot*

Description

This function generate plots for the diggit conditional analysis

Usage

```
## S4 method for signature 'diggit'
plot(x, mr = NULL, cluster = NULL, sub = NULL, ...)
```

Arguments

x	Diggit class object
mr	Optional vector of character strings indicating the MR names
cluster	Optional vector of cluster names
sub	Optional sub-title for the plot
...	Additional parameters to pass to the plot function

Value

Nothing, plots are generated in the default output device

Examples

```
data(gbm.expression, package="diggitdata")
data(gbm.cnv, package="diggitdata")
data(gbm.aracne, package="diggitdata")
dobj <- diggitClass(expset=gbmExprs, cnv=gbmCNV, regulon=gbmTFregulon)
dobj <- fCNV(dobj)
dobj <- aqtl(dobj, mr=c("CEBPD", "STAT3"), fcnv.adjust="fdr", verbose=FALSE)
dobj <- conditional(dobj, pheno="subtype", group1="MES", group2="PN", mr="STAT3", verbose=FALSE)
plot(dobj, cluster="3")
```

print,diggit-method *Basic methods for class diggit*

Description

This document lists a series of basic methods for the class diggit

Usage

```
## S4 method for signature 'diggit'
print(x, pval = 0.05)

## S4 method for signature 'diggit'
show(object)

## S4 method for signature 'diggit'
exprs(object)

## S4 method for signature 'diggit'
diggitCNV(x)

## S4 method for signature 'diggit'
diggitRegulon(x)
```

```
## S4 method for signature 'diggit'
diggitMindy(x)

## S4 method for signature 'diggit'
diggitFcnv(x)

## S4 method for signature 'diggit'
diggitMR(x)

## S4 method for signature 'diggit'
diggitViper(x)

## S4 method for signature 'diggit'
diggitAqtl(x)

## S4 method for signature 'diggit'
diggitConditional(x)

## S4 method for signature 'diggit'
summary(object)

## S4 method for signature 'diggit'
head(x, rows = 4, cols = 4)

## S4 method for signature 'diggit'
mindyFiltering(x, mr = 0.01, mr.adjust = c("none", "fdr",
  "bonferroni"))
```

Arguments

`x`	Object of class diggit
`pval`	P-value threshold for the conditional analysis
`object`	Object of class diggit
`rows`	Integer indicating the maximum number of rows to show
`cols`	Integer indicating the maximum number of columns to show
`mr`	Either a numerical value between 0 and 1 indicating the p-value threshold for the Master Regulator (MR) selection, or a vector of character strings listing the MRs
`mr.adjust`	Character string indicating the multiple hypothesis test correction for the MRs

Value

print returns summary information about the diggit object

show returns summary information about the object of class diggit

exprs returns the ExpressionSet object containing the expression profile data

diggitCNV returns a matrix containing the CNV data diggitRegulon returns a regulon object containing the transcriptional interactome

diggitMindy returns a regulon object containing the post-translational interactome

diggitFcnv returns a vector of p-values for the F-CNVs

diggitMR returns a vector of master regulators NES

diggitViper returns a matrix of VIPER results

diggitAqtl returns a matrix of aQTLs (p-value)

diggitConditional returns a list containing the conditional analysis results

summary returns the integrated results from the conditional analysis

head returns a list containing a reduced view for an object of class diggit

mindyFiltering returns a diggit class object with CNV and aQTL slots filtered to contain only MINDy post-translational modulators of the MRs

Examples

```
data(gbm.expression, package="diggitdata")
data(gbm.cnv, package="diggitdata")
data(gbm.aracne, package="diggitdata")
dobj <- diggitClass(expset=gbmExprs, cnv=gbmCNV, regulon=gbmTFregulon)
print(dobj)
show(dobj)
exprs(dobj)
diggitCNV(dobj)[1:3, 1:3]
diggitRegulon(dobj)
diggitMindy(dobj)
diggitFcnv(dobj)
diggitMR(dobj)
diggitViper(dobj)
diggitAqtl(dobj)
diggitConditional(dobj)
head(dobj)
data(gbm.expression, package="diggitdata")
data(gbm.cnv, package="diggitdata")
data(gbm.mindy, package="diggitdata")
dobj <- diggitClass(expset=gbmExprs, cnv=gbmCNV, mindy=gbmMindy)
dobj <- fCNV(dobj)
dobj
dobj <- mindyFiltering(dobj, mr=c("STAT3", "CEBPD"))
dobj
```

Index