Package: cancerGI (via r-universe)

October 23, 2024

Type Package

Title Analyses of Cancer Gene Interaction

Version 1.0.1

Date 2023-08-30

Author Audrey Qiuyan Fu and Xiaoyue Wang

Maintainer Audrey Q. Fu <audreyqyfu@gmail.com>

Description Functions to perform the following analyses: i) inferring epistasis from RNAi double knockdown data; ii) identifying gene pairs of multiple mutation patterns; iii) assessing association between gene pairs and survival; and iv) calculating the smallworldness of a graph (e.g., a gene interaction network). Data and analyses are described in Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. <doi:10.1038/ncomms5828>.

Depends R (>= 2.10)

License GPL (>= 2)

LazyLoad yes

LazyData yes

Imports systemfit, qvalue, survival, reshape2, igraph

NeedsCompilation no

Date/Publication 2023-09-07 06:52:33 UTC

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

RemoteUrl https://github.com/cran/cancerGI

RemoteRef HEAD

RemoteSha 6ec6ae0c5bffb1d9d1e5a319b9ad5c808b28324e

Contents

`computeSmallWorldness` *Compute smallworldness of a graph*

Description

This function computes the smallworldness of a graph.

Usage

```
computeSmallWorldness(g, n, m, nrep = 1000)
```

Arguments

`g`	A graph object.
`n`	Number of nodes of g.
`m`	Number of edges of g.
`nrep`	Number of random graphs to generate for estimating C_{rand} and L_{rand}.

Details

For a graph g with n nodes and m edges, the smallworldness S is defined as in Humphries and Gurney (2008):

$S = (C_g/C_{rand})/(L_g/L_{rand})$,

where C_g and C_{rand} are the clustering coefficient of g and that of a random graph with the same number of nodes and edges as g, respectively. Also, L_g and L_{rand} are the mean shortest path length of g and that of the same random graph, respectively.

Here, in order to estimate C_{rand} and L_{rand}, this function generates a large number of random graphs with n nodes and m edges under the Erdos-Renyi model (Erdos and Renyi, 1959), such that each edge is created with the same probability as the nodes in g. This function then computes C and L for each random graph, and takes the average as the estimate for C_{rand} and L_{rand}.

Value

A scalar of smallworldness.

Author(s)

Audrey Q. Fu

References

Humphries, M. D. and Gurney, K. Network 'small-world-ness': a quantitative method for determining canonical network equivalence. PLoS ONE 3, e0002051 (2008).

Erdos, P. and Renyi, A. On random graphs. Publ. Math. 6, 290-297 (1959).

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

Examples

```
library (igraph)
# compute smallworldness for the design graph
data (tested_pairs)
# build the graph object
g <- graph.edgelist (as.matrix (tested_pairs), directed=FALSE)
summary (g)  # 67 nodes and 1508 edges
# compute smallworldness
computeSmallWorldness (g, n=67, m=1508)
```

---

computeSurvivalPValueForGenePairSet.output

Survival analysis for pairs of genes

Description

This function counts the number of individuals with different mutation patterns, estimates the median survival time for each mutation pattern, and computes the p values.

Usage

```
computeSurvivalPValueForGenePairSet.output(file.out,
 gene.pairs, data.mut, data.surv,
 colTime = 2, colStatus = 3,
 type.gene1 = (-1), type.gene2 = (-1),
 groups = c("All", "Two"),
 PRINT = FALSE, PRINT.INDEX = FALSE)
```

Arguments

file.out	Output filename.
gene.pairs	Matrix of two columns, which are gene names.
data.mut	Integer matrix of genes by cases. The first column contains gene names. Each of the other columns contains mutation patterns of a case: 0 as wildtype, 1 amplification and -1 deletion.
data.surv	Data frame containing case ID, survival time and survival status. Cases do not need to match those in data.mut.
colTime	Scalar indicating which column in data.surv contains the survival time.
colStatus	A character string indicating which column in data.surv contains the survival status: "DECEASED" or "LIVING".
type.gene1	Integer indicating the type of mutation: 0 for wild type, 1 for amplification, and -1 for deletion.
type.gene2	Same as type.gene1, but for the second gene.
groups	"All" if comparing all combinations: wildtype & wildtype, wild type & mutated, both mutated; or "Two", if only comparing single mutation and double mutation.
PRINT	Default is FALSE. Prints intermediate values if set to TRUE. Output may be massive if the number of gene pairs is large.
PRINT.INDEX	Default is FALSE. Unused.

Value

Data frame containing the following columns (if groups="Two"):

gene1	
gene2	
nSingleMut	No. of cases with single mutation
nDoubleMut	No. of cases with double mutation
obsSingleMut	No. of deceased cases with single mutation
obsDoubleMut	No. of deceased cases with double mutation
expSingleMut	Expected no. of deceased cases with single mutation
expDbouleMut	Expected no. of deceased cases with double mutation
medianSingleMut	Estimated median survival time for single mutation
medianDoubleMut	Estimated median survival time for double mutation
pValue	p value for testing whether double/single mutation is associated with survival

Author(s)

Audrey Q. Fu, Xiaoyue Wang

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

Examples

```
## Not run:
data (mutations)
data (survival)

# compute p values for gene pairs tested in the RNAi knockdown assay
data (tested_pairs)

# compute p values for the gain & loss combination
# and compare only cases of single mutations with cases of double mutations;
# results are written to file tmp.txt under current directory
computeSurvivalPValueForGenePairSet.output (file.out="tmp.txt",
 tested_pairs, data.mut=mutations, data.surv=survival,
 type.gene1=1, type.gene2=(-1), groups="Two")


## End(Not run)
```

---

computeSurvivalPValueGenePairAll.output

Survival analysis for pairs of genes (with matched individuals)

Description

This function is similar to `computeSurvivalPValueForGenePairSet.output`, except that individuals in `data.mut` and `data.surv` should match, and that `gene.pairs` contains four columns: gene1, mutation type of gene1, gene2, mutation type of gene2.

Usage

```
computeSurvivalPValueGenePairAll.output(file.out,
 gene.pairs, data.mut, data.surv,
 colTime = 2, colStatus = 3,
 groups = c("All", "Two"),
 PRINT = FALSE, PRINT.INDEX = FALSE)
```

Arguments

`file.out`	Output filename.
`gene.pairs`	Matrix of four columns: gene1, mutation type of gene1, gene2, mutation type of gene2.

data.mut	Integer matrix of genes by cases. The first column contains gene names. Each of the other columns contains mutation patterns of a case: 0 as wildtype, 1 amplification and -1 deletion.
data.surv	Data frame containing case ID, survival time and survival status. Cases should match those in data.mut.
colTime	Scalar indicating which column in data.surv contains the survival time.
colStatus	A character string indicating which column in data.surv contains the survival status: "DECEASED" or "LIVING".
groups	"All" if comparing all combinations: wildtype & wildtype, wild type & mutated, both mutated; or "Two", if only comparing single mutation and double mutation.
PRINT	Default is FALSE. Prints intermediate values if set to TRUE. Output may be massive if the number of gene pairs is large.
PRINT.INDEX	Default is FALSE. Unused.

Author(s)

Audrey Q. Fu, Xiaoyue Wang

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

See Also

Called by computeSurvivalPValueForGenePairSet.output

---

computeSurvivalPValueOneGenePair

Survival analysis for one pair of genes

Description

This function performs survival analysis, similar to function computeSurvivalPValueForGenePairSet.output, but for one pair of genes.

Usage

```
computeSurvivalPValueOneGenePair(data.mut, data.surv,
 colTime = 2, colStatus = 3,
 type.gene1 = (-1), type.gene2 = (-1),
 groups = c("All", "Two"),
 compare = c("Both", "Gene1", "Gene2"),
 PLOT = FALSE, PRINT = FALSE,
 pvalue.text.x = 10, pvalue.text.y = 0.1,
 legend.x = 150, legend.y = 1)
```

Arguments

data.mut	Integer matrix of individuals by two genes. Each column containing the mutation patterns of multiple genes: 0 as wildtype, 1 amplification and -1 deletion.
data.surv	Data frame containing case ID, survival time and survival status. Cases should match those in data.mut.
colTime	Scalar indicating which column in data.surv contains the survival time.
colStatus	A character string indicating which column in data.surv contains the survival status: "DECEASED" or "LIVING".
type.gene1	Integer indicating the type of mutation: 0 for wild type, 1 for amplification, and -1 for deletion.
type.gene2	Same as type.gene1, but for the second gene.
groups	"All" if comparing all combinations: wildtype & wildtype, wild type & mutated, both mutated; or "Two", if only comparing single mutation and double mutation.
compare	"Both" if comparing all four combinations: wildtype & wildtype, wildtype & mutated, mutated & wildtype, and mutated & mutated. "Gene1" if comparing three combinations: gene1 wildtype, gene1 mutated & gene2 wildtype, and both mutated. "Gene2" is similar to "Gene1".
PLOT	If TRUE, plot the survival curves and print the p value onto the plot. Location of the p value legend is controlled by pvalue.text.x and pvalue.text.y described below.
PRINT	If TRUE, print intermediate values.
pvalue.text.x	The x coordinate of the p value legend in plot.
pvalue.text.y	The y coordinate of the p value legend in plot.
legend.x	The x coordinate of the curve legend in plot.
legend.y	The y coordinate of the curve legend in plot.

Value

The output contains the same info as described in computeSurvivalPValueForGenePairSet.output.

Author(s)

Audrey Q. Fu, Xiaoyue Wang

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

See Also

computeSurvivalPValueForGenePairSet.output

computeSurvivalPValueOneGenePair.output
Write results from survival analysis to output for one pair of genes

Description

This function is similar to computeSurvivalPValueOneGenePair, except that it writes the analysis results directly to output file and does not allow for plotting the survival curves.

Usage

```
computeSurvivalPValueOneGenePair.output(file.out, genes.info,
 data.mut, data.surv, colTime = 2, colStatus = 3,
 groups = c("All", "Two"), PRINT = FALSE)
```

Arguments

file.out	Output filename.
genes.info	A vector of 6 elements: gene1, mutation type, gene2, mutation type, gene1's column index in data.mut, gene2's column index in data.mut.
data.mut	Integer matrix of genes by cases. The first column contains gene names. Each of the other columns contains mutation patterns of a case: 0 as wildtype, 1 amplification and -1 deletion.
data.surv	Data frame containing case ID, survival time and survival status. Cases should match those in data.mut.
colTime	Scalar indicating which column in data.surv contains the survival time.
colStatus	A character string indicating which column in data.surv contains the survival status: "DECEASED" or "LIVING".
groups	"All" if comparing all combinations: wildtype & wildtype, wild type & mutated, both mutated; or "Two", if only comparing single mutation and double mutation.
PRINT	Default is FALSE. Prints intermediate values if set to TRUE. Output may be massive if the number of gene pairs is large.

Value

A vector of values from the survival analysis, as described in computeSurvivalPValueForGenePairSet.output

Author(s)

Audrey Q. Fu, Xiaoyue Wang

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

See Also

computeSurvivalPValueForGenePairSet.output

---

constructDesignMatrix *Generate a design matrix from raw RNAi data.*

Description

This function takes the raw RNAi data as input and generates a design matrix for regression. Specifically written for the format of the data set RNAi, which contains four batches. This R function will use batch3 as the baseline.

Usage

```
constructDesignMatrix(data, covariates)
```

Arguments

data	Matrix of RNAi measurements; includes columns batch, query_gene and template_gene.
covariates	Vector of strings; each string is the name of a covariate.

Value

A design matrix. The number of rows is the same as that of the data set RNAi, and the number of columns is the same as the length of covariates.

Examples

```
## See example in documentation for the data set RNAi.
```

---

mutations *Genetic mutation data in patients.*

Description

Data frame that contains mutation patterns in multiple genes across multiple patients.

Format

A data frame with 85 rows and 951 columns. Each row is a gene. The first column contains gene names, and each of the other columns contains the mutation pattern in an individual: 0 for no mutation, 1 amplification and -1 deletion.

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828.

Examples

```
data(mutations)
```

`processDataMutSurv` *Find matched individuals in mutation and survival data*

Description

This functions finds matched individuals in `data.mut` and `data.surv`, and outputs the two data sets with only matched individuals.

Usage

```
processDataMutSurv(data.mut, data.surv, colTime = 2, colStatus = 3)
```

Arguments

`data.mut`	Integer matrix of genes by cases. The first column contains gene names. Each of the other columns contains mutation patterns of a case: 0 as wildtype, 1 amplification and -1 deletion.
`data.surv`	Data frame containing case ID, survival time and survival status. Cases do not need to match those in `data.mut`.
`colTime`	Scalar indicating which column in `data.surv` contains the survival time.
`colStatus`	A character string indicating which column in `data.surv` contains the survival status: "DECEASED" or "LIVING".

Value

A list of two data frames, `data.mut` and `data.surv`. Format of the data frames is the same as input, except that the individuals in the two data frames are matched.

Author(s)

Audrey Q. Fu, Xiaoyue Wang

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

See Also

`computeSurvivalPValueForGenePairSet.output`

RNAi *Molecular phenotypes from single and double knockdowns in RNAi screen*

Description

Single and double siRNA knockdowns were performed for genes and gene pairs. Multiple molecular phenotypes, such as the number of cells, cell size, nucleus size, etc., were measured.

Value

A data matrix with each row a knockdown experiment.

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

Examples

```
## Not run:
library (systemfit)
library (qvalue)

data (RNAi)
data (tested_pairs) # gene pairs tested in the RNAi knockdown assay

# extract gene names and put in a vector
genelist <- union(unique(RNAi$template_gene),unique(RNAi$query_gene))
genelist <- genelist[!((genelist=="empty")|(genelist=="NT"))]

# create the interaction terms for linear model
sorted_tested_pairs <- apply(tested_pairs,1,
 function(x){if (x[1]>x[2]) return (c(x[2],x[1]))
 else return(c(x[1],x[2]))})
pairs_names <- apply(sorted_tested_pairs,2,
 function(x) {paste(x[1],x[2],sep=":")})

# create vector of covariates
# using batch3 as baseline
regressors <- c("batch1","batch2","batch4",genelist,pairs_names)

# construct the design matrix
my_matrix=constructDesignMatrix(data=RNAi, covariates=regressors)

# n (cell number) and csize (cell size) are on log2 scale already
# need to transform nsize (nucleus size) to original scale
RNAi.tmp <- RNAi
RNAi$nsize <- 2^RNAi.tmp$nsize
rm (RNAi.tmp)
```

```
# create formula from column names
# using all columns
#eqlog2n <- as.formula (paste ("RNAi$n ~ ",
# paste (colnames (my_matrix), collapse="+"), sep=''))
#eqlog2csize <- as.formula (paste ("RNAi$csize ~ ",
# paste (colnames (my_matrix), collapse="+"), sep=''))
#eqnsize <- as.formula (paste ("RNAi$nsize ~ ",
# paste (colnames (my_matrix), collapse="+"), sep=''))

# test run with the first 500 columns
eqlog2n <- as.formula (paste ("RNAi$n ~ ",
 paste (colnames (my_matrix)[1:500], collapse="+"), sep=''))
eqlog2csize <- as.formula (paste ("RNAi$csize ~ ",
 paste (colnames (my_matrix)[1:500], collapse="+"), sep=''))
eqnsize <- as.formula (paste ("RNAi$nsize ~ ",
 paste (colnames (my_matrix)[1:500], collapse="+"), sep=''))

system <- list (cell.number = eqlog2n, cell.size = eqlog2csize, nuc.size=eqnsize)

# perform seemingly unrelated regression
fitsur <- systemfit (system, "SUR", data=cbind (RNAi, my_matrix), maxit=100)

# extract coefficient estimates
log2n_fitsur_coef <- coef (summary (fitsur$eq[[1]]))
log2csize_fitsur_coef <- coef (summary (fitsur$eq[[2]]))
nsize_fitsur_coef <- coef (summary (fitsur$eq[[3]]))

# compute q values
log2n_coef_q <- qvalue (log2n_fitsur_coef[,4])$qvalues
log2csize_coef_q <- qvalue (log2csize_fitsur_coef[,4])$qvalues
nsize_coef_q <- qvalue (nsize_fitsur_coef[,4])$qvalues

# build three matrices of results
log2n_fitsur_coef <- data.frame (log2n_fitsur_coef, qvalue=log2n_coef_q)
colnames (log2n_fitsur_coef) <- c("Estimate", "StdError", "tValue", "pValue", "qValue")
dim (log2n_fitsur_coef)
head (log2n_fitsur_coef)

log2csize_fitsur_coef <- data.frame (log2csize_fitsur_coef, qvalue=log2csize_coef_q)
colnames (log2csize_fitsur_coef) <- c("Estimate", "StdError", "tValue", "pValue", "qValue")
dim (log2csize_fitsur_coef)
head (log2csize_fitsur_coef)

nsize_fitsur_coef <- data.frame (nsize_fitsur_coef, qvalue=nsize_coef_q)
colnames (nsize_fitsur_coef) <- c("Estimate", "StdError", "tValue", "pValue", "qValue")
dim (nsize_fitsur_coef)
head (nsize_fitsur_coef)

## End(Not run)
```

survival *Patient survival data.*

Description

Data set that contains the survival time (in months), survival status and other information of patients.

Format

A data frame with 950 observations on the following 5 variables.

`CaseID` A vector of character strings

`OverallSurvivalMonths` A numeric vector

`OverallSurvivalStatus` A factor with levels `DECEASED` `LIVING`

`MutationCount` A numeric vector

`FractionOfCopyNumberAlteredGenome` A numeric vector

Source

Data were downloaded from http://www.cbioportal.org/.

References

Data were described and analyzed in Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828.

Examples

```
data(survival)
```

tested_pairs *Gene pairs tested in the double knockdown assay.*

Description

It contains two columns of gene names.

Format

A data frame with 1508 observations on the following 2 variables.

`V1` a character vector

`V2` a character vector

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

Examples

```
data(tested_pairs)
## see documentation for dataset \code{\link{RNAi}}
```

`testMutationalPatternAll.wrapper`
Compute the p and q values of all pairwise gene mutation patterns

---

Description

This function computes the p and q values of all pairwise gene mutation patterns. Patterns include both genes losing their function, one gene gaining function and the other losing function, both genes gaining function, and the two genes being mutually exclusive.

Usage

```
testMutationalPatternAll.wrapper(data, QVALUE = TRUE, PRINT = FALSE)
```

Arguments

`data`	Matrix of gene mutations. Each row is a gene. The first column contains gene names, and all the other columns each contain mutation values in an individual. Value 1 corresponds to gain of function, -1 loss of function, and 0 no change. Missing values are denoted NAs.
`QVALUE`	TRUE if q values are calculated, and FALSE otherwise.
`PRINT`	TRUE if printing intermediate values, and FALSE otherwise.

Value

A list of two matrices, one containing the p values, and the other the q values (if the QVALUE argument set to TRUE). Each matrix has the following columns: gene 1, gene 2, p (or q) value of the loss & loss, gain & loss, loss & gain, gain & gain, and mutually exclusive combination.

Author(s)

Audrey Fu, Xiaoyue Wang

References

Wang, X., Fu, A. Q., McNerney, M. and White, K. P. (2014). Widespread genetic epistasis among breast cancer genes. Nature Communications. 5 4828. doi: 10.1038/ncomms5828

Examples

```
data (mutations)
mut.pqvalues <- testMutationalPatternAll.wrapper (data=mutations, QVALUE=TRUE)
summary (mut.pqvalues)
dim (mut.pqvalues$pvalues)
dim (mut.pqvalues$qvalues)
mut.pqvalues$pvalues[1:10,]
```

Index