Package ‘MICMIC’

March 18, 2017

Type Package

Title Methylation regulation network inference with Conditional Mutual information based PC-algorithem

Version 0.99.0

Author Tong Yin

biocViews GeneRegulation

Maintainer Tong Yin <tongyin9002@gmail.com>

Description MICMIC is an information theory based package to infer the methylation involved regulation network. It contains some tools to 1. measure the mutual information and conditional mutual information between continuous variables; 2. construct direct correlation network; 3. study the methylation regulation for target genes within a given range on the genome.

License GPL (>=3)

LazyData TRUE

Imports MASS (>= 7.3), parallel, ggplot2, gridExtra, methods, stats, utils, cubature

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`add_edge` *add_edge*

Description

This function is to Create a method to add edges

Usage

```
add_edge(object, vertex_pairs)
```

Arguments

`object,`	Object of class Network.
`vertex_pairs,`	numeric matrix that store the pairs of vertexs

Value

a new object that store the network which added edges

Author(s)

Tong Yin

CMI *Conditional Mutual Information*

Description

`CMI` takes three continuous variables as input and calculate the conditional mutual information between X and Y based on the condition of Z. Different from estimator method based on data discretization, this fucntion will use covarians transformation to estimate the continuous probabilities distribution of x and y values.

Usage

```
CMI(X,Y,Z,method=c("covariance"),unit=c("bits",
"nats","hartley","normalized"),pvalue=FALSE,permutation_times=100)
```

Arguments

`X`	a numeric vector to test
`Y`	a numeric vector to test
`Z`	a numeric vector as the condition
`method`	the estimator method to test the CMI: "covariance"
`unit`	The unit of the result: "bits", "nats", "hartley" and "normalized" (the default is "bits"). The normalized result will be between 0 and 1.
`pvalue`	a logical value to determine whether to calculate the pvalue or not
`permutation_times`	integral value to determin the permutation times in calculating p value.

Value

a numeric value of conditional mutual information between X and Y based on condition of Z

Author(s)

Tong Yin

References

Zhang, X. (2011). Inferring gene regulatory networkds from gene expression data by path consistency algorithm based on conditional mutual information

Pethel, S.D. and Hahs, D.W. (2014). Exact Test of Independence Using Mutual Information

Examples

```
x<-rnorm(100)
y<-0.7*x+rnorm(100,sd=0.1)
z<-0.8*x+rnorm(100,sd=0.1)
cor(x,y);cor(x,y);cor(y,z)  #correlation test cannot identify the direct connection
CMI(x,y,z)  # CMI identify the direct connection between x and y is not relying on
            #  the condiction of z
CMI(y,z,x)  # CMI identify the direct connection between y and z is not relying on
```

```
           #  the condiction of x
CMI(x,z,y)  # CMI identify the connection between x and z is depending on the condition of y
CMI(x,y,z,pvalue=TRUE)$adj.pvalue
CMI(x,z,y,pvalue=TRUE)$adj.pvalue
```

CMI_met_cis_network *Conditional mutual information learning the methylation cis-acting regulation network*

Description

This function is to infer the cis-acting regulatory network between DNA methylation and gene expression

Usage

```
CMI_met_cis_network(met_data_matrix,exp_data_matrix,gene_list,distance=300000,
ref_gene_bed,ref_CpGs_bed,outfiledir=NA,pvalue_cut=0.001,core_num=1,permutation_times=100)
```

Arguments

`met_data_matrix`	a numeric matrix containing CpGs methylation data where columns contain samples and rows contain variables(probe site)
`exp_data_matrix`	a numeric matrix containing gene expression data where columns contain samples and rows contain variables(gene site)
`gene_list`	a vector containing the names of target genes
`distance`	integer specifying the upstream/downstream genome range to be analyzed
`ref_gene_bed`	a data.frame containing reference gene coorinate with five columns named "name", "chr", "start", "end" and "strand". The coordinates of genes in exp_data_matrix are required to be included in this data.frame.
`ref_CpGs_bed`	a data.frame containing reference CpGS coorinate with four columns names "name", "chr", "start" and "end". The coordinates of CpGs/probes in met_data_matrix are required to be included in this data.frame.
`outfiledir`	a string of file directory to store the result files. If the parameter is not specified, the log file directory will be get by `getwd()`.
`pvalue_cut`	the cutoff of pvalue. The default is 0.01.
`core_num`	the cpu number using for parallel computation in PC_para
`permutation_times`	the number of times of permutation to calculate the pvalue

Value

the adjacency matrix of the network with value of 0 and 1. 1 means that there is an edge between the rowname and colname of the element. And 0 means there is no edge.

Author(s)

Tong Yin

Examples

```
data("TCGA_STAD_data")
gene_name<-"MLH1"
## Not run:
network<-CMI_met_cis_network(met_data_matrix=STAD_met_data_matrix,
exp_data_matrix=STAD_exp_data_matrix,gene_list=gene_name,distance=300000,
ref_gene_bed=STAD_ref_gene_bed,ref_CpGs_bed=STAD_ref_CpGs_bed,pvalue_cut=0.00001,
permutation_times=20)

## End(Not run)
```

`delete_edge` *delete_edge*

Description

This function is to Create a method to delete edges

Usage

```
delete_edge(object, vertex_pairs)
```

Arguments

`object,`	Object of class Network.
`vertex_pairs,`	numeric matrix that store the pairs of vertexs

Value

a new object that store the network which deleted edges

Author(s)

Tong Yin

`entropy` *entropy*

Description

`entropy` takes discrete or continuous as input and calculate the entropy of X or joint entropy of X and Y.

Usage

```
entropy(X,Y=NULL,method=c("covariance","density"),
unit=c("bits","nats","hartley"),variable=c("continuous","discrete"))
```

Arguments

`X`	a numeric vector to test
`Y`	a numeric vector to test, default is NULL. If Y is given, then the joint entropy of X and Y will be calculated.
`method`	the method to estimate the probability distribution: "covariance" or "density" method. The covariance method uses equation covariance matrix which was described by Zhang, X in 2012. And the density method use the `density()` and `kde2d()` function to estimate the variables' density.
`unit`	The unit of the result: "bits", "nats", "hartley" (the default is "bits").
`variable`	variable type: "continuous" or "discrete"

Value

a numeric value of entropy

Author(s)

Tong Yin

References

Zhang, X., Zhao, X. M., He, K., Lu, L., Cao, Y., Liu, J., ... & Chen, L. (2012). Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. Bioinformatics, 28(1), 98-104.

Moon, Y. I., Rajagopalan, B., & Lall, U. (1995). Estimation of mutual information using kernel density estimators. Physical Review E, 52(3), 2318.

Venables, W. N., & Ripley, B. D. (2013). Modern applied statistics with S-PLUS. Springer Science & Business Media.

Examples

```
x1<-rnorm(100,mean=50,sd=16);x2<-c(1:100);x3<-c(1:100)+rnorm(100)
entropy(x1)
entropy(x2)
entropy(x3)
entropy(X=x1,Y=x3)
```

`generate_regulator_info` *generate_regulator_info*

Description

This function is to integrate the regulator information in the gene_regulator_info.txt file

Usage

```
generate_regulator_info(met_data_matrix,exp_data_matrix,gene_list,
outfiledir=NA,ref_gene_bed,ref_CpGs_bed)
```

Arguments

`met_data_matrix`	a numeric matrix containing CpGs methylation data where columns contain samples and rows contain variables(probe site)
`exp_data_matrix`	a numeric matrix containing gene expression data where columns contain samples and rows contain variables(gene site)
`gene_list`	a vector containing the names of target genes
`outfiledir`	a string of file directory to store the result files. If the parameter is not specified, the log file directory will be get by `getwd()`.
`ref_gene_bed`	a data.frame containing reference gene coorinate with five columns named "name", "chr", "start", "end" and "strand". The coordinates of genes in exp_data_matrix are required to be included in this data.frame.
`ref_CpGs_bed`	a data.frame containing reference CpGS coorinate with four columns names "name", "chr", "start" and "end". The coordinates of CpGs/probes in met_data_matrix are required to be included in this data.frame.

Value

data.frame containing information of direct and indirect regulators

Author(s)

Tong Yin

Examples

```
data("TCGA_STAD_data")
gene_name<-"MLH1"
## Not run:
generate_regulator_info(met_data_matrix=STAD_met_data_matrix,
exp_data_matrix=STAD_exp_data_matrix,gene_list=gene_name,
ref_gene_bed=STAD_ref_gene_bed,ref_CpGs_bed=STAD_ref_CpGs_bed)

## End(Not run)
```

`get_edge_number` *get_edge_number*

Description

This function is to get number of edges

Usage

```
get_edge_number(object)
```

Arguments

`object,`	Object of class Network.

Value

number

Author(s)

Tong Yin

`get_nearest_elements` *get_nearest_elements This function is to get the neighbour elements on genome for target gene*

Description

get_nearest_elements This function is to get the neighbour elements on genome for target gene

Usage

```
get_nearest_elements(gene_list, distance = 3e+05, gene_number = 100,
  ref_gene_bed, ref_CpGs_bed, data_node_list)
```

Arguments

`gene_list`	a character vector containing the list of target genes which to be tested
`distance`	a numeric value determining the genome range of potential cis-acting network
`gene_number`	a numeric value determining the max number of neighbour genes in this range
`ref_gene_bed`	a data.frame containing reference gene coorinate with five columns named "name", "chr", "start", "end" and "strand". The coordinates of genes in exp_data_matrix are required to be included in this data.frame.
`ref_CpGs_bed`	a data.frame containing reference CpGS coorinate with four columns names "name", "chr", "start" and "end". The coordinates of CpGs/probes in met_data_matrix are required to be included in this data.frame.
`data_node_list`	a list of node names in data matrix. The return elements which are not in the node list will be excluded.

Value

a data frame containing the neighbour genes and neighbour CpGs

Author(s)

Tong Yin

`get_partners` *get_partners*

Description

This function is to Create a method to get partners for any vertex

Usage

```
get_partners(object, vertex)
```

Arguments

`object,`	Object of class Network.
`vertex,`	a vector that store the names of nodes

Value

a vector of partners

Author(s)

Tong Yin

`get_sharing_partners` *get_sharing_partners*

Description

This function is to Create a method to get sharing partners for vertex

Usage

```
get_sharing_partners(object, vertex1, vertex2)
```

Arguments

`object,`	Object of class Network.
`vertex1,`	vector that store the name of the first node
`vertex2,`	vector that store the name of the secode node

Value

a vector of partners

Author(s)

Tong Yin

`get_vertex_number` *get_vertex_number*

Description

This function is to get number of vertexs

Usage

```
get_vertex_number(object)
```

Arguments

`object,`	Object of class Network.

Value

number

Author(s)

Tong Yin

`HNSC_control_id` *the sample ids of normal samples in TCGA HNSC dataset*

Description

A vector containing sample ids

Usage

```
data(TCGA_HNSC_data)
```

Format

A vector of character

Value

A vector of sample ids

Source

`http://cancergenome.nih.gov/`

References

Cancer Genome Atlas Network. "Comprehensive genomic characterization of head and neck squamous cell carcinomas." Nature 517.7536 (2015): 576-582.

`HNSC_exp_data_matrix` *expression values of genes in HNSC samples*

Description

A data matrix containing the log2 transformed expression levels of genes in head and neck cancer samples from TCGA

Usage

```
data(TCGA_HNSC_data)
```

Format

A matrix of numeric values

Value

The log2 expression matrix for head and neck cancer samples

Source

`http://cancergenome.nih.gov/`

References

Cancer Genome Atlas Network. "Comprehensive genomic characterization of head and neck squamous cell carcinomas." Nature 517.7536 (2015): 576-582.

`HNSC_met_data_matrix` *methylation values of CpGs in HNSC samples*

Description

A data matrix containing the methylation beta values of CpGs in head and neck cancer samples from TCGA

Usage

```
data(TCGA_HNSC_data)
```

Format

A matrix of numeric values

Value

The methylation beta matrix for head and neck cancer samples

Source

http://cancergenome.nih.gov/

References

Cancer Genome Atlas Network. "Comprehensive genomic characterization of head and neck squamous cell carcinomas." Nature 517.7536 (2015): 576-582.

`HNSC_ref_CpGs_bed` *the CpGs coordinates in bed format*

Description

A data.frame containing the chromosome coordinate information of all the CpGs nearby the target genes

Usage

```
data(TCGA_HNSC_data)
```

Format

A data.frame

Value

A data.frame containing genome coordinates of CpGs

name name of CpGs

chr the chromosome id like chr1, chr2, chr3 ...

start the starting coordinate of CpGs

end the ending coordinate of CpGs

Source

https://genome.ucsc.edu

`HNSC_ref_gene_bed` *the gene coordinates in bed format*

Description

A data.frame containing the chromosome coordinate information of all the genes nearby the target genes

Usage

```
data(TCGA_HNSC_data)
```

Format

A data.frame

Value

A data.frame containing genome coordinates of genes

name name of genes

chr the chromosome id like chr1, chr2, chr3 ...

start the starting coordinate of genes

end the ending coordinate of genes

strand the strand of genes

Source

https://genome.ucsc.edu

`HNSC_sample_class` *the classes of samples*

Description

A data.frame containing the classes(control,tumor,stages or subtypes) information of samples

Usage

```
data(TCGA_HNSC_data)
```

Format

A data.frame

Value

sample classification information in two column data.frame

sample_id name of samples, should be exactly the same as the colnames of data matrix

class control, tumor, stages, or tumor subtypes

Source

https://genome.ucsc.edu

`HNSC_tumor_id` *the sample ids of tumor samples in TCGA HNSC dataset*

Description

A vector containing sample ids

Usage

```
data(TCGA_HNSC_data)
```

Format

A vector of character

Value

A vector of sample ids

Source

http://cancergenome.nih.gov/

References

Cancer Genome Atlas Network. "Comprehensive genomic characterization of head and neck squamous cell carcinomas." Nature 517.7536 (2015): 576-582.

`merge_regulator_info` *merge_regulator_info*

Description

This function is to merge regulation information for multiple genes

Usage

```
merge_regulator_info(gene_list,outfiledir=NA,statisticfiledir=NA,ref_gene_bed)
```

Arguments

`gene_list`	a vector containing the names of target genes
`outfiledir`	a string of file directory to store the result files. If the parameter is not specified, the log file directory will be get by `getwd()`.
`statisticfiledir`	summary directory to store merged result. If the parameter is not specified, the file directory will be get by `getwd()`.
`ref_gene_bed`	a data.frame containing reference gene coorinate with five columns named "name", "chr", "start", "end" and "strand". The coordinates of genes in exp_data_matrix are required to be included in this data.frame.

Value

numbers of direct and indirect regulators

Author(s)

Tong Yin

Examples

```
## Not run:
data("TCGA_LUAD_data")
gene_list<-rownames(LUAD_exp_data_matrix)[41:50]

network<-CMI_met_cis_network(met_data_matrix=LUAD_met_data_matrix,
exp_data_matrix=LUAD_exp_data_matrix,gene_list=gene_list,distance=300000,
ref_gene_bed=LUAD_ref_gene_bed,ref_CpGs_bed=LUAD_ref_CpGs_bed,
core_num=1,permutation_times=20)

generate_regulator_info(met_data_matrix=LUAD_met_data_matrix,
exp_data_matrix=LUAD_exp_data_matrix,gene_list=gene_list,
ref_gene_bed=LUAD_ref_gene_bed,ref_CpGs_bed=LUAD_ref_CpGs_bed)

merge_regulator_info(gene_list=gene_list,ref_gene_bed=LUAD_ref_gene_bed)

## End(Not run)
```

`MI` *Mutual Informaiton*

Description

`MI` takes two continuous variables as input and calculate the mutual information between them in various units. Different from estimator method based on data discretization, this fucntion will use covarians transformation or density estimation to estimate the continuous probabilities distribution of x and y values.

Usage

```
MI(X,Y,method=c("covariance","KDE"),unit=c("bits",
"nats","hartley","normalized"),pvalue=FALSE,permutation_times=100)
```

Arguments

`X`	a numeric vector to test
`Y`	a numeric vector to test
`method`	choose an estimator method to test the mutual information: "covariance" or "KDE" (the default is "covariance").
`unit`	The unit of the result: "bits", "nats", "hartley" and "normalized" (the default is "bits"). The normalized result will be between 0 and 1.
`pvalue`	a logical value to determine whether to calculate the pvalue or not
`permutation_times`	integral value to determin the permutation times in calculating p value.

Value

a numeric value of mutual information between X and Y

Author(s)

Tong Yin

Examples

```
x=rnorm(100);y1=rnorm(100);y2=x+rnorm(100)
MI(x,y1)
MI(x,y2)
MI(x,y2,pvalue=TRUE)
```

MICMIC_plotting *MICMIC_plotting*

Description

This function is to map genome coordinates to plotting coordinates, and enlarge target gene promoter and gene body

Usage

```
MICMIC_plotting(gene_name,met_data_matrix,exp_data_matrix,control_id,
distance=NA,ref_gene_bed,ref_CpGs_bed,sample_class,outfiledir=NA)
```

Arguments

gene_name	The name of target gene to be plotted
met_data_matrix	a numeric matrix containing CpGs methylation data where columns contain samples and rows contain variables(probe site)
exp_data_matrix	a numeric matrix containing gene expression data where columns contain samples and rows contain variables(gene site)
control_id	a vector containing the ids of control/normal samples
distance	Integer specifying the upstream/downstream genome range to be plotted. By default distance will cover all CpGs in analysis result
ref_gene_bed	a data.frame containing reference gene coorinate with five columns named "name", "chr", "start", "end" and "strand". The coordinates of genes in exp_data_matrix are required to be included in this data.frame.
ref_CpGs_bed	a data.frame containing reference CpGS coorinate with four columns names "name", "chr", "start" and "end". The coordinates of CpGs/probes in met_data_matrix are required to be included in this data.frame.
sample_class	a data.frame containing the class information for samples
outfiledir	a string of file directory to store the result files. If the parameter is not specified, the log file directory will be get by `getwd()`.

Value

a ggplot object

Author(s)

Tong Yin

Examples

```
data("TCGA_STAD_data")
gene_name<-"MLH1"
## Not run:
MICMIC_plotting(gene_name=gene_name,met_data_matrix=STAD_met_data_matrix,
exp_data_matrix=STAD_exp_data_matrix,control_id=STAD_control_id,
distance=350000,ref_gene_bed=STAD_ref_gene_bed,
ref_CpGs_bed=STAD_ref_CpGs_bed,sample_class=sample_class)

## End(Not run)
```

`Network` *Network An S4 class to store the network in adjacent matrix*

Description

Network An S4 class to store the network in adjacent matrix

Slots

`vertex`, a string vector to store the node names

`edges`, a numeric matrix with two columns to store the edges

`adj_matrix`, a adjacent matrix to store the unidirectional network

`bi_adj_matrix`, a adjacent matrix to sotre the bidirectional network

Author(s)

Tong Yin

`PC_para` *parallel PC network construction based on MI/CMI testing*

Description

`PC_para` is a parallel computation method to infer direct correlation network from data matrix. This method is based on PC-algorithm by conditional mutual information It will generate an adjacent matrix of the infered network.

Usage

```
PC_para(data_matrix,max_L=1,method=c("CMII","CMI"),pre_adj=NULL
,log_file_dir=NA,edgemode=c("pvalue"),pvalue_cut=0.01,core_num=1,permutation_times=100)
```

Arguments

`data_matrix`	a numeric data matrix containing data from observation where columns contain samples(observing) and rows contain variables
`max_L`	The max L of PC. The default value is 1, and that means the network will be infered by CMI testing. If the value is 0, the network will be infered by MI testing.
`method`	choose a to test interaction between nodes based on conditional mutual information (CMI), or conditional mutual inclusive information.
`pre_adj`	the pre-defined adjacent matrix, representing the hypothetical network. The default value is NULL, and that means all nodes are considered to have association between each other in original hypothesis.
`log_file_dir`	a string of file directory to store the log files. If the parameter is not specified, the log file directory will be get by `getwd()`.
`edgemode`	a string value to select the mode in edge decision
`pvalue_cut`	the cutoff of pvalue. The default is 0.01.
`core_num`	the number of CPUs using in the computation.
`permutation_times`	the number of times of permutation to calculate the pvalue

Value

the adjacency matrix of the network with value of 0 and 1. 1 means that there is an edge between the rowname and colname of the element. And 0 means there is no edge.

Author(s)

Tong Yin

References

Zhang, X. (2011). Inferring gene regulatory networkds from gene expression data by path consistency algorithm based on conditional mutual information

Zhang, X. (2015). Conditional mutual inclusive information enables accurate quatification of associations in gene regulatory networks.

Kalisch, M. and Buhlmann, P.(2007) Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm.

Pethel, S.D. and Hahs, D.W. (2014). Exact Test of Independence Using Mutual Information

Examples

```
x=rnorm(300,mean=20,sd=6)
y=x+rnorm(300,mean=0,sd=2)
w=y*0.1+rnorm(300,mean=18,sd=1)
v=y*0.15+rnorm(300,mean=17,sd=1)
z=2*w+v+rnorm(300,mean=0,sd=0.1)
a=rnorm(300,mean=20,sd=2)
b=0.9*a+rnorm(300,mean=2,sd=1)
c=b-rnorm(300,mean=0,sd=2)
mydata<-rbind(x,y,w,v,z,a,b,c)
MI_PC_net<-PC_para(mydata,max_L=0)
CMI_PC_net<-PC_para(mydata,max_L=1)
```

`read_adj_matrix` *read_adj_matrix*

Description

This function is to Create a method to read genes annotation data

Usage

```
read_adj_matrix(object, adj_matrix)
```

Arguments

`object,`	Object of class Network.
`adj_matrix,`	an adjacent matrix to load

Value

the new object loaded with adjacent matrix

Author(s)

Tong Yin

`STAD_control_id` *the sample ids of normal samples in TCGA STAD dataset*

Description

A vector containing sample ids

Usage

```
data(TCGA_STAD_data)
```

Format

A vector of character

Value

A vector of sample ids

Source

`http://cancergenome.nih.gov/`

References

Cancer Genome Atlas Research Network. "Comprehensive molecular characterization of gastric adenocarcinoma." Nature 513.7517 (2014): 202-209.

`STAD_exp_data_matrix` *expression values of genes in STAD samples*

Description

A data matrix containing the log2 transformed expression levels of genes in gastric cancer samples from TCGA

Usage

```
data(TCGA_STAD_data)
```

Format

A matrix of numeric values

Value

The log2 expression matrix for gastric cancer samples

Source

`http://cancergenome.nih.gov/`

References

Cancer Genome Atlas Research Network. "Comprehensive molecular characterization of gastric adenocarcinoma." Nature 513.7517 (2014): 202-209.

`STAD_met_data_matrix` *methylation values of CpGs in STAD samples*

Description

A data matrix containing the methylation beta values of CpGs in gastric cancer samples from TCGA

Usage

```
data(TCGA_STAD_data)
```

Format

A matrix of numeric values

Value

The methylation beta matrix for gastric cancer samples

Source

`http://cancergenome.nih.gov/`

References

Cancer Genome Atlas Research Network. "Comprehensive molecular characterization of gastric adenocarcinoma." Nature 513.7517 (2014): 202-209.

`STAD_ref_CpGs_bed` *the CpGs coordinates in bed format*

Description

A data.frame containing the chromosome coordinate information of all the CpGs nearby the target genes

Usage

```
data(TCGA_STAD_data)
```

Format

A data.frame

Value

A data.frame containing genome coordinates of CpGs

name name of CpGs

chr the chromosome id like chr1, chr2, chr3 ...

start the starting coordinate of CpGs

end the ending coordinate of CpGs

Source

`https://genome.ucsc.edu`

`STAD_ref_gene_bed` *the gene coordinates in bed format*

Description

A data.frame containing the chromosome coordinate information of all the genes nearby the target genes

Usage

```
data(TCGA_STAD_data)
```

Format

A data.frame

Value

A data.frame containing genome coordinates of genes

name name of genes

chr the chromosome id like chr1, chr2, chr3 ...

start the starting coordinate of genes

end the ending coordinate of genes

strand the strand of genes

Source

`https://genome.ucsc.edu`

`STAD_sample_class` *the classes of samples*

Description

A data.frame containing the classes(control,tumor,stages or subtypes) information of samples

Usage

```
data(TCGA_STAD_data)
```

Format

A data.frame

Value

sample classification information in two column data.frame

sample_id name of samples, should be exactly the same as the colnames of data matrix

class control, tumor, stages, or tumor subtypes

Source

`https://genome.ucsc.edu`

`STAD_tumor_id` *the sample ids of tumor samples in TCGA STAD dataset*

Description

A vector containing sample ids

Usage

```
data(TCGA_STAD_data)
```

Format

A vector of character

Value

A vector of sample ids

Source

http://cancergenome.nih.gov/

References

Cancer Genome Atlas Research Network. "Comprehensive molecular characterization of gastric adenocarcinoma." Nature 513.7517 (2014): 202-209.

`test_vertex_pairs` *test_vertex_pairs*

Description

This function is to test whether the nodes in the input edges are in the network or not

Usage

```
test_vertex_pairs(object, vertex_pairs)
```

Arguments

`object,`	Object of class Network.
`vertex_pairs,`	numeric matrix that store the pairs of vertexs

Value

testing information

Author(s)

Tong Yin

update_edge *update_edge*

Description

This function is to Create a method to update edge data from object

Usage

```
update_edge(object)
```

Arguments

`object,`	Object of class Network.

Value

a new object that store the network which added edges

Author(s)

Tong Yin

Index