Package 'CAsubtype'

September 30, 2016

Type Package

Title CAsubtype: an R package to identify gene sets predictive of cancer subtypes and clinical outcomes

Version 1.1

Date 2016-09-30

Author Hualei Kong, Hua Li

Maintainer Hualei Kong<khl0798@163.com>

Description
CAsubtype is a flexible and well-integrated tool in the R environment to identify gene sets for cancer subtype identification and clinical outcome prediction.

License GPL-3

Depends R (>= 2.10), Biobase

Imports RCurl, XML (>= 2.8.0), amap, beanplot (>= 1.2), MASS, cluster (>= 2.0.3), survival (>= 2.38.3), plyr (>= 1.8.3), methods, affy, biomaRt, BiocGenerics, scatterplot3d

Collate allGenerics.R CopeMicroarray.R database.R dataClasses.R inputfile.R Method-Genesig.R Method-Msigdb.R plotfigure.R survival.R TCGA_true.r trueclustering.R

Repository CRAN

LazyLoad yes;

NeedsCompilation no

R topics documented:

`CAsubtype-package` *CAsubtype*

Description

CAsubtype: an R package to identify gene sets predictive of cancer subtypes and clinical outcomes

Details

The DESCRIPTION file: Important data classes: `GetCancer`, `GetMsigdb`, `GetGenesig`. Perform PCA analysis: `PcCompute` Perform clustering analysis: `SampleCluster` Survival analysis: `SurvivalDist`

Author(s)

Hualei Kong<khl0798@163.com>

`CancerType` *Get type of cancer in TCGA open access data*

Description

Get type of cancer in TCGA open access data.

Usage

```
CancerType()
```

Value

`character value`
short name of cancer type

Examples

```
#cancerTy=CancerType()
```

`CheckConstant` *Removing genes which have constant expressions*

Description

Check and remove genes which have constant expressions

Usage

```
CheckConstant(samples)
```

Arguments

`samples`	a data frame or matrix, the rownames are corresponding to genes, while colnames are corresponding to samples.

Value

`samples`	a data frame or matrix

Examples

```
Breast=GetCancer("Breast")

cancerSamples=CheckConstant(t(exprs(Breast)))
```

CheckPackage *Check weather package is loaded*

Description

Check weather package is loaded

Usage

```
CheckPackage(name)
```

Arguments

name	a character, the name of package

Value

TRUE or FALSE	If TRUE, package have already loaded.

Examples

```
CheckPackage("CAsubtype")
```

Clinical *Download clinical data from TCGA*

Description

Download clinical data from TCGA.

Usage

```
Clinical(cancertype,typename,organization)
```

Arguments

cancertype	short for cancer type, for example, stomach cancer can be short for "stad".
typename	default is "bcr", short for "Biospecimen Core Resource".
organization	the source of data, one of "biotab", "intgen.org", "nationwidechildrens.org".

Details

This function is used for downloading Clinical data in TCGA, and data is divided into two levels, "Level_1" and "Level_2". "Level_1" used for "intgen.org" and "nationwidechildrens.org"", while "Level_2" is only used for "nationwidechildrens.org".

Value

file	Downloaded clinical file in current working directory

Examples

```
#clinData=Clinical(cancertype="stad",typename="bcr",organization="biotab")
```

`ClusterSample` *Cluster Samples by Kmeans or Hierarchy algorithm clustering methods.*

Description

cluster samples by kmeans or hierarchy algorithm clustering methods;

Usage

```
ClusterSample(input, cluster.number, clustering.method="hierarchical",
           ifpca=TRUE, pc.num, start=2, end=5, count=20, distance="euc", dists=FALSE)
```

Arguments

`input`	a data frame
`cluster.number`	a numeric value set the specific clustering number ;
`clustering.method`	a character,'kmeans' or 'hierarchical '; default is "hierarchical".
`ifpca`	logical value, if TRUE, samples will be clustered after PCA analysis; default is TRUE.
`pc.num`	numeric value, choose defined number of first few principal components
`start`	numeric value, set minimum number of clustering; default is 2.
`end`	numeric value, set maximum number of clustering; default is 5.
`count`	numeric value, set defined number in repeatly computing silhouette; default is 20.
`distance`	a character, use 'euclidean' distance or 'correlation' to cluster, if the 'euclidean' method is choosen, the minimum number of components is 2; if the 'correlation' method is choosen,the minimum number of components is 3; short for 'euc','cor' seperately; default is "euc".
`dists`	a logical value, cluster samples according to 'dist' or 'dissimilarity' class. if FALSE distance is "euc", otherwise distance is "cor" ; default is FALSE.

Value

`clust.result`	a list which involving *clustgroup*, a data frame, the index of samples referring to gene signature; *sample.pc.data*, a list, the first few principal components of samples in gene signature

See Also

amap,SampleCluster

Examples

```
sample=matrix(rnorm(5000,0,10), ncol=50, byrow=TRUE)

results=ClusterSample(input=sample,cluster.number=5, pc.num=3)
```

CopeMicroarray *Cope with microarray data*

Description

CopeMicroarray is used for process raw expressions of probe ids and convert probe ids into gene symbols, removing duplicated genes.

Usage

```
CopeMicroarray(arrayfile,affyprobeDataSet,DatasetName="hsapiens_gene_ensembl",arryattributes="hg
```

Arguments

arrayfile	'CEL' files stored the raw expressions of probe id in the current working directory.
affyprobeDataSet	this argument is the same as attributes in getBM function, represent the short name of microarray which produced the probe id.
DatasetName	default is "hsapiens_gene_ensembl".
arryattributes	default is "hgnc_symbol".

Details

microarray is used for 'Hsapiens' by default.

Value

a data frame	the gene expression profiles

See Also

See Also as UpData, getBM, useMart, listDatasets, useDataset

DescriptionClinical *Brief description of Clinical information in TCGA*

Description

Brief description of Clinical information in TCGA

Usage

```
DescriptionClinical()
```

Value

character string	Information of Clinical outcomes on cancers in TCGA

Examples

```
#DescriptionClinical()
```

DescriptionDownloadedGeneExp *Description of gene expression profiles in TCGA*

Description

Description of gene expression profiles in TCGA

Usage

```
DescriptionDownloadedGeneExp()
```

Value

`character string` Information of Gene expression profiles on cancers in TCGA

Examples

```
#DescriptionDownloadedGeneExp()
```

DownloadClinicalSample *A direct R command to download and process clinical data in TCGA*

Description

A direct R command to download and process clinical data in TCGA

Usage

```
DownloadClinicalSample(cancertype,typename)
```

Arguments

`cancertype`	short for cancer type, for example, stomach cancer can be short for "stad"
`typename`	default is "bcr", short for "Biospecimen Core Resource",

Details

`DownloadClinicalSample` function provide users automatically to download and process the clinical data from TCGA

Value

`a list`	3 elements : biotab, intgen, nation; clinical data in 3 organization platform

Examples

```
#clinicaldata=DownloadClinicalSample(cancertype="stad",typename="bcr")
```

DownloadGeneExp *Download gene expression profile from TCGA*

Description

Download gene expression profile from TCGA.

Usage

```
DownloadGeneExp(platform,cancertype,organization,typename,RPKM)
```

Arguments

platform	Platform which is used for producing gene expression profiles; when organization is "unc.edu", it corresponding to 'illuminahiseq'; 'illuminaga' or 'illuminahiseq' for 'bcgsc.ca'
cancertype	short for cancer type, for example, stomach cancer can be short for "stad"
organization	the source of data, one of "unc.edu", "bcgsc.ca"
typename	default is "cgcc"
RPKM	a logical value, there are two methods to compute the gene expressions, if FALSE, it will be RSEM. if organization is "unc.edu", it will be FALSE, while "bcgsc.ca" it will be TRUE.

Details

The RPKM is used for extracting data which is computed by RPKM or RSEM.

Value

a list or data frame
: gene expression profiles

Examples

```
#dlgenedata=DownloadGeneExp(platform="illuminaga",cancertype="stad",organization="bcgsc.ca",typename="cgcc"
```

DrugResponse *Drug responses on cancers*

Description

The DrugResponse data is corresponding to gene expression profiles get by GetCancer function.

Usage

```
DrugResponse
```

Format

a matrix, 201 type of drugs and 1654 cancer samples, While rows stand for drugs.

Examples

```
#data(DrugResponse)
```

EMT *Genes in EMT gene signature*

Description

The EMT including genes in EMT gene signature.

Usage

```
EMT
```

Format

a list includes 310 number of genes

Examples

```
#data(EMT)
```

Genesig-class *Class* "Genesig"

Description

Container for gene signatures in GeneSigDB database

Objects from the Class

Objects can be created by calls of the form `new("Genesig", db,dbstats)`.

Slots

`db` "data.frame"

`symbol` a list, gene symbols in each gene signature

Methods

show `object`Brief description of GeneSigDB database

GeneSigDBDescription `GeneSigDBDescription(object)`Brief description of GenesigDB database

GeneSigDBGeneNum `GeneSigDBGeneNum(object)`Number of genes in gene signature

GeneSigDBTissue `GeneSigDBTissue(object,GName)`Tissues or organs produced gene signature

GeneSigDBGPMID `GeneSigDBPMID(object,GName)`PMID of assay contained gene signature

GeneSigGeneSymbols `GeneSigGeneSymbols(object,GName)`Gene symbols in gene signature

GeneSigDBOrganism `GeneSigDBOrganism(object,GName)`Species produced the gene signature

GeneSigDBPlatform `GeneSigDBPlatform(object,GName)`Platforms produced the gene signature

GeneSigDBPlatformDescription `GeneSigDBPlatformAccesionID(object,GName)`Brief description of platforms which produced gene signature

GeneSigDBSigName `GeneSigDBSigName(object,GName)`Names of gene signatures

GeneSigDBGeneSymbolDescription `GeneSigDBGeneSymbolDescription(object,GName)`Brief functional description of gene signature

References

http://genesigdb.org/genesigdb/

`GeneSigDBDescription` *Description of GeneSigDB database*

Description

Brief Description of GeneSigDB database

Usage

```
GeneSigDBDescription(object)
```

Arguments

`object`	a "Genesig" S4 class

Examples

```
DlGene=GetGenesig(species="Human")
GeneSigDBDescription(DlGene)
```

`GeneSigDBGeneNum` *Sizes of gene signatures in GenesigDB database*

Description

Sizes of gene signatures in GenesigDB database

Usage

```
GeneSigDBGeneNum(object,GName)
```

Arguments

`object`	a "Genesig" S4 class
`GName`	names of gene signatures

Value

`numeric value` Sizes of gene signatures in GenesigDB database

Examples

```
DlGene=GetGenesig(species="Human")
geneNum=GeneSigDBGeneNum(DlGene,rownames(DlGene@db)[1:10])
```

`GeneSigDBGeneSymbolDescription`
Functional description of gene signatures in GenesigDB database

---

Description

Functional description of gene signatures in GenesigDB database

Usage

```
GeneSigDBGeneSymbolDescription(object,GName)
```

Arguments

`object`	a "Genesig" S4 class
`GName`	names of gene signatures

Value

`a character value`
Functional description of gene signatures , if not exists, return NULL.

See Also

`Genesig-class`

Examples

```
DlGene=GetGenesig(species="Human")
geneDes=GeneSigDBGeneSymbolDescription(DlGene,rownames(DlGene@db)[1:10])
```

GeneSigDBOrganism *Organisms produced the gene signatures in GenesigDB database*

Description

Organisms produced the gene signatures in GenesigDB database

Usage

```
GeneSigDBOrganism(object,GName)
```

Arguments

object	a "Genesig" S4 class
GName	names of gene signatures

Value

a character value
Organisms of gene signatures, if not exists, return NULL.

See Also

Genesig-class

Examples

```
DlGene=GetGenesig(species="Human")
geneOrganism=GeneSigDBOrganism(DlGene,rownames(DlGene@db)[1:10])
```

GeneSigDBPlatform *Search for the Platform of gene signatures in GenesigDB database*

Description

search for the Platform of gene signatures in GenesigDB database

Usage

```
GeneSigDBPlatform(object,GName)
```

Arguments

object	a "Genesig" S4 class
GName	names of gene signatures

Value

a character value
Platform of gene signatures, if not exists, return NULL.

See Also

Genesig-class

Examples

```
DlGene=GetGenesig(species="Human")
platform=GeneSigDBPlatform(DlGene,rownames(DlGene@db)[1:10])
```

GeneSigDBPlatformDescription *Brief description of Platforms which produced gene signatures in GenesigDB database*

Description

Brief description of Platforms which produced gene signatures in GenesigDB database

Usage

```
GeneSigDBPlatformDescription(object,GName)
```

Arguments

object	a "Genesig" S4 class
GName	names of gene signatures

Value

a character value
: Brief description of Platforms which produced gene signatures in GenesigDB database, if not exists, return NULL.

See Also

Genesig-class

Examples

```
#DlGene=GetGenesig(species="Human")
#geneNum=GeneSigDBPlatformDescription(DlGene,rownames(DlGene@db)[1:10])
```

GeneSigDBPMID *PMID index used for extracting Gene Signatures in GenesigDB database*

Description

PMID index used for extracting Gene Signatures in GenesigDB database

Usage

```
GeneSigDBPMID(object, GName)
```

Arguments

object	a "Genesig" S4 class
GName	names of gene signatures

Value

character value or NULL

PMID index used for extracting Gene Signatures in GenesigDB database

Examples

```
DlGene=GetGenesig(species="Human")
gpmid=GeneSigDBPMID(DlGene,rownames(DlGene@db)[1:10])
```

GeneSigDBTissue *Tissues used for producing gene signatures in GenesigDB database*

Description

Tissues used for producing gene signatures in GenesigDB database

Usage

```
GeneSigDBTissue(object,GName)
```

Arguments

object	a "Genesig" S4 class
GName	names of gene signatures

Value

a character value

Tissues used for producing gene signatures in GenesigDB database, if not exists, return NULL.

See Also

Genesig-class

Examples

```
DlGene=GetGenesig(species="Human")
GeneTissue=GeneSigDBTissue(DlGene,rownames(DlGene@db)[1:10])
```

GeneSigGeneSymbols *Extract Gene symbols of gene signatures from GeneSigDB database*

Description

Extract Gene symbols of gene signatures from GeneSigDB database

Usage

```
GeneSigGeneSymbols(object,GName)
```

Arguments

object	a "Genesig" S4 class
GName	names of gene signatures

Value

a character value

Extract Gene symbols of gene signatures from GeneSigDB database, if not exists, return NULL.

See Also

Genesig-class

Examples

```
DlGene=GetGenesig(species="Human")
GeneSymbol=GeneSigGeneSymbols(DlGene,rownames(DlGene@db)[1:10])
```

GetCancer *Get the gene expressions profiles and clinical information of cancer from CAsubtype package*

Description

Get the gene expressions profiles and clinical information of cancer from CAsubtype package.

Usage

```
GetCancer(cancertype)
```

Arguments

cancertype	a character value, involving "Bladder ; Breast ; Colon ; Brain ; Head and Neck ; Kidney ; Lung ; Ovary ; Rectum ; Endometrial ; Gastric ;"

Value

a s4 class	an "ExpressionSet" class, includes gene expression profiles and patients' clinical information ;

Note

11 cancer types (Bladder ; Breast ; Colon ; Brain ; Head and Neck ; Kidney ; Lung ; Ovary ; Rectum ; Endometrial ; Gastric) of gene expression profiles and clinical information can get from CAsubtype package.

References

Bioconductor: Open software development for computational biology and bioinformatics. R. Gentleman, V. J. Carey, D. M. Bates,B.Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, and others , Genome Biology, Vol. 5, R80. 2004 .

See Also

ExpressionSet

Examples

```
Breast=GetCancer("Breast")
```

GetGenesig *Retrieve gene signatures of GeneSigDB database from CAsubtype package. (GetGenesig)*

Description

Retrieve gene signatures of GeneSigDB database from CAsubtype package.

Usage

```
GetGenesig(species)
```

Arguments

species	a character value, including "Human", "Mouse";

Value

db	a data frame involves the basic information of the gene signatures;
symbol	a list involves gene symbols of each gene siganture in GeneSigDB database

References

Culhane A C, Schwarzl T, Sultana R, et al. *GeneSigDB a curated database of gene expression signatures*. Nucleic acids research, 38, suppl 1, D716-D725. 2010.

Culhane A C, Schroder M S, Sultana R, et al.*GeneSigDB: a manually curated database and resource for analysis of gene expression signatures*. Nucleic acids research, gkr901. 2011.

Examples

```
DlGene=GetGenesig(species="Human")
```

GetMsigdb *Retrieve gene signatures of MSigDB databases from CAsubtype package*

Description

Retrieve gene signatures of MSigDB databases from CAsubtype package

Usage

```
GetMsigdb(species, categorycode="all")
```

Arguments

species	a character value, including "Human", "Mus musculus", "Rattus norvegicus"
categorycode	a character vector, contains "H", from "C1" to "C8", "all", default is "all";

Value

This is a S4 class .

STANDARD_NAME	Character value which contains the standard names of gene signatures
SYSTEMATIC_NAME	Character value which contains the systematic names
HISTORICAL_NAMES	Character value which contains the historical names
ORGANISM	Character value which contains the species
PMID	Numeric value which contains the PUBMED which the gene signatures were developed
AUTHORS	Character value, the author who develop the gene signatures
GEOID	Character value which is the GEO id of gene signatures
EXACT_SOURCE	Character value
GENESET_LISTING_URL	Character value - URL
EXTERNAL_DETAILS_URL	Character value
CHIP	Character value
CATEGORY_CODE	Character value, the divided 9 major collections, "H", from "C1" to "C8", default is "all"
SUB_CATEGORY_CODE	Character value
CONTRIBUTOR	Character value
CONTRIBUTOR_ORG	Character value
DESCRIPTION_BRIEF	Briefly description of gene signatures
DESCRIPTION_FULL	Fully description of gene signatures
TAGS	character value
MEMBERS	Gene symbols
MEMBERS_SYMBOLIZED	Gene symbols
MEMBERS_EZID	Entriz id of genes
MEMBERS_MAPPING	Character value
FOUNDER_NAMES	Character value
REFINEMENT_DATASETS	Character value
VALIDATION_DATASETS	Character value

References

Liberzon A, Subramanian A, Pinchback R, Thorvaldsdottir H, Tamayo P, Mesirov JP. *Molecular signatures database (MSigDB) 3.0.* Bioinformatics. Jun 15; 27(12), 1739-40. Epub 2011 May 5. 2011.

See Also

http://software.broadinstitute.org/gsea/msigdb/index.jsp.

Examples

```
msigdb=GetMsigdb(species="Human")
```

`geturlTCGA` *Parse XML format to nodes on websites of TCGA access data*

Description

Parse XML format to nodes on websites of TCGA access data

Usage

```
geturlTCGA(object)
```

Arguments

`object`	website address from TCGA

Value

`character value`

Deparsed nodes in TCGA access data website

Note

`geturlTCGA`

Examples

```
#DataUrl=geturlTCGA("https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/")
```

`IntegratedClinical` *Download clinical outcomes of cancers directly from TCGA*

Description

Download clinical outcomes of cancers directly from TCGA

Usage

```
IntegratedClinical(object,organization)
```

Arguments

`object`	working directory that stored the downloaded clinical data
`organization`	the original sources of data, "biotab", "intgen.org", "nationwidechildrens.org".

Value

`a list` clinical outcomes

See Also

`Clinical`, `DownloadClinicalSample`

Examples

```
#library(XML,quietly=TRUE)
#library(RCurl,quietly=TRUE)
#clinData=Clinical(cancertype="stad",typename="bcr",organization="biotab")
#workdir=file.path(getwd(),paste("Clinical","stad","bcr","biotab",sep="_"),"/")
#DownloadClin=IntegratedClinical(object=workdir,organization="biotab")
```

Msigdb-class *Class* "Msigdb"

Description

Container for gene signatures in MsigDB database.

Objects from the Class

Objects can be created by calls of the form `new("Msigdb",STANDARD_NAME, SYSTEMATIC_NAME, HISTORICAL_NAMES,`

Slots

STANDARD_NAME Character value involves standard names of gene signatures in MSigDB database

SYSTEMATIC_NAME Character value involves systematic names of gene signatures in MSigDB database

HISTORICAL_NAMES Character value involves historical names of gene signatures in MSigDB database

ORGANISM Character value of the species produced gene signatures

PMID Character values contains the PUBMED id where gene signatures extracted

AUTHORS Character value of the author who developed gene signatures

GEOID Character value point to GEO id where gene signatures developed

EXACT_SOURCE Character value

GENESET_LISTING_URL Character value - URL

EXTERNAL_DETAILS_URL Character value

CHIP Character value

CATEGORY_CODE Character value, categorycode-"H", from "C1" to "C8", default is "all"

SUB_CATEGORY_CODE Character value

CONTRIBUTOR Character value

CONTRIBUTOR_ORG Character value

DESCRIPTION_BRIEF Briefly description of gene signatures

DESCRIPTION_FULL Fullt description of the gene signatures

TAGS Character value

MEMBERS Gene symbols

MEMBERS_SYMBOLIZED Gene symbols

MEMBERS_EZID Entriz id of genes

MEMBERS_MAPPING Character value

FOUNDER_NAMES Character value

REFINEMENT_DATASETS Character value

VALIDATION_DATASETS Character value

Methods

show `object`Brief description of `Msigdb` function

MSigDBDescription `MSigDBDescription(object)`Brief Description of MsigDB database

MSigDBPMID `MSigDBPMID(object,Name)`Extract the PMID of gene signature

MSigDBGeneSymbol `MSigDBGeneSymbol(object,Mname)`Extract the gene symbols in gene signature

MSigDBGeneNum `MSigDBGeneNum(object)`The number of genes in gene signature

MSigDBDescriptionGeneSymbol `MSigDBDescriptionGeneSymbol(object,Mname)`Brief description of function in gene signature

References

http://software.broadinstitute.org/gsea/msigdb/index.jsp

`MSigDBDescription` *Brief description of MsigDB database*

Description

Brief description of MsigDB database.

Usage

```
MSigDBDescription(object)
```

Arguments

`object`	a "Msigdb" S4 class

Examples

```
DlMsig=GetMsigdb(species="Human",categorycode="C1")
MSigDBDescription(DlMsig)
```

MSigDBDescriptionGeneSymbol *Functional description of gene signatures in MSigDB database*

Description

Functional description of gene signatures in MSigDB database

Usage

```
MSigDBDescriptionGeneSymbol(object,Mname)
```

Arguments

object	an "Msigdb" S4 class
Mname	name of Gene Signature in MsigDB database

Value

a data frame	Functional description of gene signatures in MSigDB database

See Also

Msigdb-class

Examples

```
DlMsig=GetMsigdb(species="Human",categorycode="C1")
DesGeneSignature=MSigDBDescriptionGeneSymbol(DlMsig,DlMsig@STANDARD_NAME[1:10])
```

MSigDBGeneNum *Sizes of gene signatures in MsigDB database*

Description

Sizes of gene signatures in MsigDB database

Usage

```
MSigDBGeneNum(object)
```

Arguments

object	a "Msigdb" S4 class

Value

numeric value	Sizes of gene signatures in MsigDB database

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")
MsigNumber=MSigDBGeneNum(dlMsig)
```

MSigDBGeneSymbol *Gene symbols of gene signatures in MsigDB database*

Description

Gene symbols of gene signatures in MsigDB database.

Usage

```
MSigDBGeneSymbol(object,Mname)
```

Arguments

object	a "Msigdb" S4 class
Mname	name of gene signatures in MsigDB database

Value

a list	gene symbol in each gene signature

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")
MsigGeneSym=MSigDBGeneSymbol(dlMsig,dlMsig@STANDARD_NAME[10])
```

MSigDBPMID *PMID index used for extracting Gene Signatures in MSigDB database*

Description

PMID index used for extracting Gene Signatures in MSigDB database

Usage

```
MSigDBPMID(object,Mname)
```

Arguments

object	a "Msigdb" S4 class
Mname	name of gene signatures in MsigDB database

Value

a data frame or NULL	PMID index used for extracting Gene Signatures in MSigDB database,if not exist, return NULL

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")
Msigpmid=MSigDBPMID(dlMsig,dlMsig@STANDARD_NAME[5000])
```

PcCompute *Perform Principal component analysis*

Description

Perform principal component analysis.

Usage

```
PcCompute(SampleSource, GeneSet, pc.num=3,verbose=TRUE)
```

Arguments

SampleSource	an "ExpressionSet" S4 class involves gene expression profiles and clinical information.
GeneSet	a list involves genes in gene signature
pc.num	a numberic value to choose the number of first few principal components, default is 3.
verbose	a logical value shows working procedure ;if TRUE, progress of PCA analysis applying to percentage of 25th, 50th, 75th, 100th of gene signatures will be output, default is TRUE;

Details

Perform PCA for the samples.

Value

PCvarpct	numeric value , variance percentages of the seleted principal components
Cronbach	numeric value ,Cronbach's alpha used to test if the first seleted principal components are concordance.
genes	genes contained in gene signatures
num	a numeric value, sizes of gene signatures

References

R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/. 2015.

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")

Breast=GetCancer("Breast")

BreastPC=PcCompute(SampleSource=Breast,GeneSet=dlMsig@MEMBERS_SYMBOLIZED[1:5])
```

PcPlot *Scatter plot of variance percentages on first few principal components vs size of gene signatures.*

Description

Scatter plot of variance percentages on first few principal components vs size of gene signatures

Usage

```
PcPlot(Sample, Sig = FALSE, siggenenames, Figname = NULL)
```

Arguments

Sample	a list returned from PcCompute function.
Sig	logical value about weather or not pointing specific gene signatures.
siggenenames	character values, if Sig is TRUE, name of gene signatures
Figname	name of figure, default is NULL

Details

PcPlot is used to draw the scatter figure on distribution of variance percentages in first few principal components vs size of gene signatures

Value

matrix	a matrix which have two columns corresponding to variance percentages of first selected gene signatures and size of gene signatures

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")

Breast=GetCancer("Breast")

BreastPC=PcCompute(SampleSource=Breast,GeneSet=dlMsig@MEMBERS_SYMBOLIZED[1:100])

PcPlot(Sample=BreastPC)
##
PcPlot(Sample=BreastPC,Sig=TRUE,siggenenames=c("chr5q23","chr16q24"))
```

PCScatter *Scatter plot of samples in two or three dimensions on first few principal components.*

Description

2 or 3 dimensions of scatter plot on cancer samples' according to first few principal components.

Usage

```
PCScatter(Sample, SetSelect = NULL, dimension=3, Figname = NULL)
```

Arguments

Sample	a list, returned from *SampleCluster* function.
SetSelect	numeric value, default is NULL, if not, number of selecting gene signatures.
dimension	a numberic value, 2 or 3, draw the scatter figures in 2 or 3 dimensions; default is 3.
Figname	name of figure, default is NULL.

Details

Input files should be returned from *ClusterSample* function.

Value

scatter figure
: Figure is produced in the current working directory names "SampleScatter"

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")

Breast=GetCancer("Breast")

BreastPC=PcCompute(SampleSource=Breast,GeneSet=dlMsig@MEMBERS_SYMBOLIZED[1:100])

SampleCLS=SampleCluster(GeneSet=BreastPC$genes, pc.num=3, SampleSource=Breast, cluster.number=3)

PCScatter(Sample=SampleCLS,SetSelect=c(1,2,3))
```

PvaluePlot *Figure of beanplot shows distribution of p-values in real gene signatures compared with simulated ones;*

Description

Figure of beanplot shows distribution of p-values in real gene signatures compared with simulated ones;

Usage

```
PvaluePlot(pvaluedata, alpha=0.05,  Figname = NULL)
```

Arguments

pvaluedata	a list returned from SurvivalSimulation function
alpha	threshold used for selecting the significant gene sets; default is 0.05.
Figname	name of figure, default is NULL.

Value

a figure	distribution of p values on survival analysis in real gene signatures compared with simulated ones.

See Also

See Also as beanplot, SurvivalDist, SurvivalSimulation

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")

Breast=GetCancer("Breast")

BreastPC=PcCompute(SampleSource=Breast,GeneSet=dlMsig@MEMBERS_SYMBOLIZED[1:10])

BreastSim=SigSimPC(SampleSource=Breast,SamplePC=BreastPC)

SampleCLS=SampleCluster(GeneSet=BreastSim$GeneSignature, pc.num=3, SampleSource=Breast,cluster.number=4)

SigP=SurvivalDist(groups=SampleCLS$clust_group, clin=phenoData(Breast)@data)

SampleSimulate=SurvivalSimulation(GeneSet=BreastSim$GeneSignature, SampleSource=Breast, Pcut=0.05,
                                  samplePvalue=SigP,cluster.number=4,pc.num=3)

SigGene=PvaluePlot(SampleSimulate)
```

SampleCluster *Cluster samples by kmeans or hierarchy algorithm clustering methods;*

Description

Cluster samples by kmeans or hierarchy algorithm clustering methods;

Usage

```
SampleCluster(SampleSource, GeneSet, pc.num=3, ifpca=TRUE,  clustering.method="hierarchical",
           distance="euc", start=2, end=5, count=20, dists=FALSE, cluster.number=0,
              wrap = FALSE, simu = FALSE)
```

Arguments

SampleSource	an "ExpressionSet" S4 class involves gene expression profiles and clinical information.
GeneSet	a list of gene signatures involves gene symbol and names of gene signatures
pc.num	the number of principal components;default is 3
ifpca	logical value weather or not choosing PCA method; default is TRUE
clustering.method	character value, "kmeans" or "hierarchy algorithm" clustering methods; default is "hierarchical".
distance	character value, using "euclidean" or "correlation" to compute the distance of samples; default is "euc".
start	if clustering.number is 0, minimum number of subgroups to find the optimum number for clustering according to "silhouette" measurements,default is 2
end	if clustering.number is 0, maximum number of subgroups to find the optimum number for clustering according to "silhouette" measurements, default is 5
count	if clustering.number is 0, repeated times to find the optimum number for clustering according to "silhouette" measurements, default is 20.
dists	a logical value, FALSE if distance is "euclidean", otherwise it will be TRUE if distance is "correlation",; default is FALSE.
cluster.number	a numeric value set the specific clustering number ; default is 0;if cluster.number is 0,use "silhouette" measurements to find the optimum number for subcluster;
wrap	a logical value, if TRUE, output records of processing samples on the gene signatures; default is FALSE.
simu	a logical value, if TRUE, group the samples according to simulated gene signatures; default is FALSE.

Details

SampleCluster function is used for grouping samples in real or simulated gene sigantures;

Value

clust_group	a data frame about the index of samples on which group they are belonging to,rownames are corresponding to samples, while colnames are stand for gene signatures;
sample_pc	a list of samples, samples' first few principal components;

References

Antoine Lucas. *amap: Another Multidimensional Analysis Package*. R package version 0.8-14.https://CRAN.R-project.org/package=amap; 2014.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.3. 2015.

See Also

PvaluePlot

SelectPC *Select top or bottom proportions of PCvarpcts in gene signatures*

Description

Select top or bottom proportions of PCvarpcts in gene signatures

Usage

```
SelectPC(SampleSource, proportion=0.01,Sig.num=c(10,200,500),max=TRUE)
```

Arguments

SampleSource	a list returned from PcCompute function
proportion	according to the different sizes of gene numbers in gene signatures, select the top proportion PCvarpct in gene signatures; default is 0.01, select top 1 percent proportions of PCvarpcts in gene signatures according to default different intervals on different size of gene signatures;
Sig.num	the intervals are corresponding to the alpha for example, default 3 intervals, [10,200], [200,500], [500,max(size of gene signatures)];
max	a logical value, if TRUE, select the top proportion of PCvarpct in gene signatures, otherwise, select the bottom proportion of PCvarpct;

Value

a list	4 elements : "num", "PCvarpct", "Cronbach", "genes"

See Also

See Also as PcCompute

Examples

```
dlMsig=GetMsigdb(species="Human",categorycode="C1")

Breast=GetCancer("Breast")

BreastPC=PcCompute(SampleSource=Breast,GeneSet=dlMsig@MEMBERS_SYMBOLIZED[1:10])

SelectBreastPC=SelectPC(SampleSource=BreastPC)
```

`SigSimPC` *Compute the significances of PCvarpcts on real gene signatures in PCA analysis*

Description

Compute the significances of PCvarpcts on real gene signatures in PCA analysis

Usage

```
SigSimPC(SampleSource,SamplePC,seeds=123456,repeats=100,verboseprocess=TRUE, pc.num=3, proportio
```

Arguments

`SampleSource`	an "ExpressionSet" S4 class, includes gene expression profiles and clinical information of cancer.
`SamplePC`	a list returned from PcCompute or Selectsig function
`seeds`	seeds in stochastic simulation,default is 123456
`repeats`	times of repetition, default is 100
`verboseprocess`	logical value, if TRUE, print the process of simulation
`pc.num`	first few principal components, default is 3.
`proportion`	threshold of p-value to select signiciant gene signatures.

Details

The distribution of PCvarpct will be zero if the PCvarpct of input gene signature is the minimum value in simulated ones,; otherwise, this value will be proportation to ranks in simulated ones.

Value

`a list`	significant gene signatures and corresponding distributions of PCvarpcts in simulated ones

See Also

`PcCompute`, `PvaluePlot`

SurvivalDist *Perform survival analysis.*

Description

Perform survival analysis.

Usage

```
SurvivalDist(groups, clin, Figname=NULL, survivalplots=TRUE, sigverbose=FALSE)
```

Arguments

groups	a data frame or matrix ,each column represents index number of subclusters on samples.
clin	clinical information of samples, includes two columns, "OS","OScensoring" on samples. 'OS' is short for "Overall Survival" ,while "OScensoring" is short for "Overall Survival censoring".
Figname	name of figure, default is NULL.
survivalplots	a logical value, if TRUE, draw figure of "Kaplan-Meier" plot . default is TRUE.
sigverbose	a logical value,default is FALSE ;if TRUE, choose gene sets which have p values less than 0.05;

Details

SruvivalDist function is used for performing the survival analysis for samples and output the P value by log-rank test.

Value

survival plots

If survivalplots is TRUE, output the "Kaplan-Meier" figures of samples in the current working directory.

a numeric value

p values computed by log-rank test on the different subgroups of samples.

References

Therneau T. *A Package for Survival Analysis in S*. version 2.38, URL:http://CRAN.R-project.org/package=survival. 2015. Terry M. Therneau and Patricia M. Grambsch . *Modeling Survival Data: Extending the Cox Model*.Springer, New York. ISBN 0-387-98784-3. 2000.

See Also

survival, PvaluePlot

SurvivalSimulation *Compute distribution of p values on survival analysis in real gene signatures compared with simulated ones*

Description

Compute distribution of p values on survival analysis in real gene signatures compared with simulated ones

Usage

```
SurvivalSimulation(GeneSet, SampleSource,samplePvalue, pc.num=3,cluster.number=0, Figname=NULL,
repeats=1000, seeds=123456,sigverbose=FALSE, Pcut=0.05,
clustering.method="hierarchical", distance="euc", start=2, ifpca=TRUE,
end=5, count=20, dists=FALSE, survivalplots=FALSE)
```

Arguments

GeneSet	a list contains genes symbols and names of gene signatures;
SampleSource	an "ExpressionSet" S4 class, involves gene expression profiles and clinical information.
samplePvalue	p-value of survival analysis on real gene signatures, returned from SurvivalDist function
pc.num	first few principal components ,default is 3;it will be omit if verbose is FALSE;
cluster.number	a numeric value set the specific clustering number ; default is 0;if cluster.number is 0,use "silhouette" measurements to find the optimum number for subcluster;
Figname	name of figure, default is NULL
repeats	a numberic value, the number of resamples ,the default is 1000;
seeds	random seed for reproducible results, default is 123456;
sigverbose	a logical value,if TRUE it will compute and draw the Kaplan-meier figure of significant gene signatures according to threshold of p-value: 0.05, default is FALSE.
Pcut	default is 0.05, cutoff to choose gene signatures which have p values less than 0.05 for survival simulation
clustering.method	a character string involves 'kmeans' or 'hierarchical'; default is "hierarchical";
distance	character, use 'euclidean' distance or 'correlation' to cluster, if the 'euclidean' method is choosen, the minimum number of principal components is 2; if the 'correlation' method is choosen, the minimum number of principal components is 3; 'euc','cor' can be insteaded, default is "euc";
start	if clustering.number is 0, minimum number of subgroups to find the optimum number for clustering according to "silhouette" measurements,default is 2
ifpca	a logical character, if TRUE then the it will be cluster according to principal component, default is TRUE;
end	if clustering.number is 0, maximum number of subgroups to find the optimum number for clustering according to "silhouette" measurements, default is 5

count	if clustering.number is 0, repeated times to find the optimum number for clustering according to "silhouette" measurements, default is 20.
dists	a logical value, FALSE if distance is "euclidean", otherwise it will be TRUE if distance is "correlation",; default is FALSE
survivalplots	logical value, if TRUE,draw "Kaplan-Meier" survival curves, default is FALSE;

Details

SurvivalSimulation function is used for compute the random cluster

Value

p values	a data frame, number of rows are equal to simulation times, number of columns are equal to the length of GeneSet;

References

Therneau T . *A Package for Survival Analysis in S*. version 2.38, URL: http://CRAN.R-project.org/package=survival. 2015.

Terry M. Therneau and Patricia M. Grambsch . *Modeling Survival Data: Extending the Cox Model*. Springer, New York. ISBN 0-387-98784-3. 2000.

Antoine Lucas . *amap: Another Multidimensional Analysis Package*. R package version 0.8-14. https://CRAN.R-project.org/package=amap.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.3. 2015.

See Also

See Also as SurvivalDist, SampleCluster, PvaluePlot

TCGAinfo *Organization that stored the gene expression profiles and clinical information in TCGA*

Description

TCGA information-organization which produce and store the gene expression profiles and clinical information of cancer samples

Usage

```
TCGAinfo(cancertype, typename)
```

Arguments

cancertype	short for cancer type, for example, stomach cancer can be short for "stad"
typename	default is "bcr", short for "Biospecimen Core Resource",

Value

`character value`
: the source of data, such as "biotab", "intgen.org", "nationwidechildrens.org"

Examples

```
#org=TCGAinfo(cancertype="stad",typename="bcr")
```

UpData — *Pack gene expression profiles and clinical data into "ExpressionSet" class*

Description

Pack gene expression profiles and clinical data into "ExpressionSet" class

Usage

```
UpData(samples,clinical)
```

Arguments

`samples`	rownames are genes, while colnames stand for samples' id.
`clinical`	clinical outcomes of samples, and rownames are samples' id. the rownames in clinical should be corresponding to colnames in samples. Columns of 'Overal Survival' and 'Overal Survival censoring'are short for "OS","OScensoring".

Value

`a S4 class`	an ExpressionSet class

See Also

See Also as `ExpressionSet`

Examples

```
Breast=GetCancer("Breast")
BreastSample=exprs(Breast)
BreastClinical=phenoData(Breast)@data
#newBreastData=UpData(BreastSample,BreastClinical)
```

Index