

# Package ‘canceR’

April 24, 2025

**Type** Package

**Title** A Graphical User Interface for accessing and modeling the Cancer Genomics Data of MSKCC

**Version** 1.43.0

**Date** 2024-02-28

**Author** Karim Mezhoud.  
Nuclear Safety & Security Department.  
Nuclear Science Center of Tunisia.

**Maintainer** Karim Mezhoud <kmezhoud@gmail.com>

**Description** The package is user friendly interface based on the cgdsr and other modeling packages to explore, compare, and analyse all available Cancer Data (Clinical data, Gene Mutation, Gene Methylation, Gene Expression, Protein Phosphorylation, Copy Number Alteration) hosted by the Computational Biology Center at Memorial-Sloan-Kettering Cancer Center (MSKCC).

**License** GPL-2

**LazyLoad** yes

**Depends** R (>= 4.3), tcltk, cBioPortalData

**Imports** GSEABase, tkrplot, geNetClassifier, RUnit, Formula, rpart, survival, Biobase, phenoTest, circlize, plyr, tidyr, dplyr, graphics, stats, utils, grDevices, R.oo, R.methodsS3

**Suggests** testthat (>= 3.1), knitr, rmarkdown, BiocStyle

**SystemRequirements** Tktable, BWidget

**VignetteBuilder** knitr

**BugReports** <https://github.com/kmezhoud/canceR/issues>

**biocViews** GUI, GeneExpression, Clustering, GO, GeneSetEnrichment, KEGG, MultipleComparison

**RoxygenNote** 7.3.1

**Encoding** UTF-8

**git\_url** <https://git.bioconductor.org/packages/canceR>

**git\_branch** devel

**git\_last\_commit** acc6e8a

**git\_last\_commit\_date** 2025-04-15

**Repository** Bioconductor 3.22

**Date/Publication** 2025-04-24

## Contents

|                                        |    |
|----------------------------------------|----|
| about . . . . .                        | 3  |
| canceR . . . . .                       | 4  |
| canceR_Issue . . . . .                 | 4  |
| canceR_Vignette . . . . .              | 5  |
| cbind.na . . . . .                     | 5  |
| CGDS . . . . .                         | 6  |
| dialogGeneClassifier . . . . .         | 6  |
| dialoggetGeneListMSigDB . . . . .      | 7  |
| dialogMetOption . . . . .              | 8  |
| dialogMut . . . . .                    | 8  |
| dialogOptionCircos . . . . .           | 9  |
| dialogOptionGSEAlm . . . . .           | 9  |
| dialogOptionPhenoTest . . . . .        | 10 |
| dialogPlotOption_SkinCor . . . . .     | 10 |
| dialogSamplingGSEA . . . . .           | 11 |
| dialogSelectFiles_GSEA . . . . .       | 12 |
| dialogSpecificMut . . . . .            | 12 |
| dialogSummary_GSEA . . . . .           | 13 |
| displayInTable . . . . .               | 13 |
| getCases . . . . .                     | 14 |
| getCasesGenProfs . . . . .             | 15 |
| getCircos . . . . .                    | 15 |
| getClinicalDataMatrix . . . . .        | 16 |
| getClinicData_MultipleCases . . . . .  | 16 |
| getCor_ExpCNAMet . . . . .             | 17 |
| geteSet . . . . .                      | 18 |
| getFreqMutData . . . . .               | 18 |
| getGCTCLSExample . . . . .             | 19 |
| getGCT_CLSfiles . . . . .              | 19 |
| getGeneExpMatrix . . . . .             | 20 |
| getGeneList . . . . .                  | 20 |
| getGeneListExample . . . . .           | 21 |
| getGeneListFromMSigDB . . . . .        | 21 |
| getGenesClassifier . . . . .           | 22 |
| getGenesTree_MultipleCases . . . . .   | 22 |
| getGenesTree_SingleCase . . . . .      | 23 |
| getGenProfs . . . . .                  | 23 |
| getGSEAlm_Diseases . . . . .           | 24 |
| getGSEAlm_Variables . . . . .          | 24 |
| getInTable . . . . .                   | 25 |
| getListProfData . . . . .              | 25 |
| getMegaProfData . . . . .              | 26 |
| getMetDataMultipleGenes . . . . .      | 26 |
| getMSigDB . . . . .                    | 27 |
| getMSigDBExample . . . . .             | 28 |
| getMSigDBfile . . . . .                | 28 |
| getMutData . . . . .                   | 29 |
| getPhenoTest . . . . .                 | 29 |
| getProfData . . . . .                  | 30 |
| getProfilesDataMultipleGenes . . . . . | 31 |

|                                     |    |
|-------------------------------------|----|
| getProfilesDataSingleGene . . . . . | 31 |
| getSpecificMut . . . . .            | 32 |
| getSummaryGSEA . . . . .            | 32 |
| getSurvival . . . . .               | 33 |
| getTextWin . . . . .                | 33 |
| GSEA . . . . .                      | 34 |
| GSEA.Analyze.Sets . . . . .         | 34 |
| GSEA.ConsPlot . . . . .             | 35 |
| GSEA.EnrichmentScore . . . . .      | 36 |
| GSEA.EnrichmentScore2 . . . . .     | 37 |
| GSEA.Gct2Frame . . . . .            | 37 |
| GSEA.Gct2Frame2 . . . . .           | 38 |
| GSEA.GeneRanking . . . . .          | 38 |
| GSEA.HeatMapPlot . . . . .          | 39 |
| GSEA.HeatMapPlot2 . . . . .         | 40 |
| GSEA.NormalizeCols . . . . .        | 40 |
| GSEA.NormalizeRows . . . . .        | 41 |
| GSEA.ReadClsFile . . . . .          | 41 |
| GSEA.Res2Frame . . . . .            | 42 |
| GSEA.Threshold . . . . .            | 42 |
| GSEA.VarFilter . . . . .            | 43 |
| GSEA.write.gct . . . . .            | 44 |
| Match_GeneList_MSigDB . . . . .     | 44 |
| modalDialog . . . . .               | 45 |
| myGlobalEnv . . . . .               | 45 |
| OLD.GSEA.EnrichmentScore . . . . .  | 46 |
| plotModel . . . . .                 | 46 |
| plot_1Gene_2GenProfs . . . . .      | 47 |
| plot_2Genes_1GenProf . . . . .      | 47 |
| rbind.na . . . . .                  | 48 |
| Run.GSEA . . . . .                  | 48 |
| setWorkspace . . . . .              | 49 |
| test.CGDS . . . . .                 | 49 |
| testCheckedCaseGenProf . . . . .    | 50 |
| UnifyRowNames . . . . .             | 50 |

**Index****52**


---

|       |                     |
|-------|---------------------|
| about | <i>about canceR</i> |
|-------|---------------------|

---

**Description**

about canceR

**Usage**

about()

**Value**

dialog box with text

**Examples**

```
## Not run:  
about()  
  
## End(Not run)
```

---

|        |                      |
|--------|----------------------|
| canceR | <i>main function</i> |
|--------|----------------------|

---

**Description**

main function

**Usage**

canceR()

**Value**

open the starting windows with cancer studies

**Examples**

```
ENV <- new.env(parent = emptyenv())  
## Not run:  
canceR()  
  
## End(Not run)
```

---

|              |                            |
|--------------|----------------------------|
| canceR_Issue | <i>canceR Report Issue</i> |
|--------------|----------------------------|

---

**Description**

canceR Report Issue

**Usage**

canceR\_Issue()

**Value**

link to github issues

**Examples**

```
## Not run:  
canceR_Issue()  
  
## End(Not run)
```

---

|                 |                          |
|-----------------|--------------------------|
| canceR_Vignette | <i>open pdf vignette</i> |
|-----------------|--------------------------|

---

**Description**

open pdf vignette

**Usage**

```
canceR_Vignette()
```

**Value**

open pdf vignette

**Examples**

```
## Not run:  
canceR_Vignette()  
  
## End(Not run)
```

---

|          |                              |
|----------|------------------------------|
| cbind.na | <i>bind non equal column</i> |
|----------|------------------------------|

---

**Description**

bind non equal column

**Usage**

```
cbind.na(..., deparse.level = 1)
```

**Arguments**

|               |                                                                                                                                                                                                                                          |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ...           | (generalized) vectors or matrices.                                                                                                                                                                                                       |
| deparse.level | integer controlling the construction of labels in the case of non-matrix-like arguments (for the default method): deparse.level = 0 constructs no labels; the default, deparse.level = 1 or 2 constructs labels from the argument names. |

**Value**

a data frame with merged columns

**Examples**

```
## Not run:
col1 <- c("a","b","c","d")
col2 <- c("A", "B", "C")
col3 <- cbind.na(col1, col2)

## End(Not run)
```

CGDS

*CGDS connect object to cBioPortal***Description**

Creates a CGDS connection object from a CGDS endpoint URL. This object must be passed on to the methods which query the server.

**Usage**

```
CGDS(url,verbose=FALSE,ploterrmsg=' ',token=NULL)
```

**Arguments**

|            |                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------|
| url        | A CGDS URL (required).                                                                                                  |
| verbose    | A boolean variable specifying verbose output (default FALSE)                                                            |
| ploterrmsg | An optional message to display in plots if an error occurs (default "")                                                 |
| token      | An optional 'Authorization: Bearer' token to connect to cBioPortal instances that require authentication (default NULL) |

dialogGeneClassifier

*Dialogue Box for gene classifier setting: sample size and postprob threshold***Description**

Dialogue Box for gene classifier setting: sample size and postprob threshold

**Usage**

```
dialogGeneClassifier(Lchecked_Cases,entryWidth = 10,returnValOnCancel = "ID_CANCEL")
```

**Arguments**

|                   |                                        |
|-------------------|----------------------------------------|
| Lchecked_Cases    | integer with a number of checked cases |
| entryWidth        | integer default 10                     |
| returnValOnCancel | "ID_CANCEL"                            |

**Value**

a dataframe with genes classes

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/gbm_tcgaPlotTwoGenProf.rds", sep=""))
## Not run:
getGenesClassifier()
dialogGeneClassifier(1,10,returnValOnCancel = "ID_CANCEL")

## End(Not run)
```

---

dialoggetGeneListMSigDB

*Multi-select choice of gene sets from loaded MSigDB*

---

**Description**

Multi-select choice of gene sets from loaded MSigDB

**Usage**

```
dialoggetGeneListMSigDB(MSigDB)
```

**Arguments**

MSigDB                      object with MSigDB. A list of genesets

**Value**

a dataframe with genes classes

**Examples**

```
z <- 7
## Not run:
##MSigDB <- readLines(paste(.libPaths(), "/cancerR/extdata/MSigDB/c5.bp.v4.0.symbols.gmt", sep=""))
dialoggetGeneListMSigDB(MSigDB)

## End(Not run)
```

---

|                 |                                              |
|-----------------|----------------------------------------------|
| dialogMetOption | <i>Dialog Box to set methylation options</i> |
|-----------------|----------------------------------------------|

---

**Description**

Dialog Box to set methylation options

**Usage**

```
dialogMetOption(ProfData, k)
```

**Arguments**

|          |                                  |
|----------|----------------------------------|
| ProfData | adataframe with methylation data |
| k        | threshold of silencing gene 0:1  |

**Value**

a dialog box to set methylation option (threshold of silencing gene)

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getMetDataMultipleGenes()
#dialogMetOption(ProfData,0.7)

## End(Not run)
```

---

|           |                                                        |
|-----------|--------------------------------------------------------|
| dialogMut | <i>Dialog bos to set returned Mutation information</i> |
|-----------|--------------------------------------------------------|

---

**Description**

Dialog bos to set returned Mutation information

**Usage**

```
dialogMut(title, question, entryInit, entryWidth = 40,returnValOnCancel = "ID_CANCEL")
```

**Arguments**

|                   |                    |
|-------------------|--------------------|
| title             | title of the table |
| question          | question           |
| entryInit         | entryInit          |
| entryWidth        | 40                 |
| returnValOnCancel | "ID_CANCEL"        |

**Value**

a check box with mutations variables

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
dialogMut("title", "question", "entryInit", entryWidth = 40, returnValOnCancel = "ID_CANCEL")

## End(Not run)
```

---

|                    |                                      |
|--------------------|--------------------------------------|
| dialogOptionCircos | <i>Checkbox to select dimensions</i> |
|--------------------|--------------------------------------|

---

**Description**

Checkbox to select dimensions

**Usage**

```
dialogOptionCircos()
```

**Value**

a checkbox with all dimensions

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/Circos.rds", sep=""))
## Not run:
dialogOptionCircos()
#getCircos(dimension ="All")

## End(Not run)
```

---

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| dialogOptionGSEAlm | <i>Dialogbox to select variables from Clinical data</i> |
|--------------------|---------------------------------------------------------|

---

**Description**

Dialogbox to select variables from Clinical data

**Usage**

```
dialogOptionGSEAlm(k,ClinicalData)
```

**Arguments**

|              |                                   |
|--------------|-----------------------------------|
| k            | integer 1                         |
| ClinicalData | dataframe with clinical variables |

**Value**

permutaion value, p-value, coVariables

**Examples**

```
#data(ClinicalData)
## Not run:
getOptionGSEAlm()

## End(Not run)
```

---

dialogOptionPhenoTest    *Checkbox to select variables from clinical data*

---

**Description**

Checkbox to select variables from clinical data

**Usage**

```
dialogOptionPhenoTest(eSet)
```

**Arguments**

eSet                      Expression Set

**Value**

vectors: variables to test Survival status, AGE, p-value

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))
## Not run:
dialogOptionPhenoTest(ENV$eSet)

## End(Not run)
```

---

dialogPlotOption\_SkinCor  
                            *Checkbox to select variables for plotting*

---

**Description**

Checkbox to select variables for plotting

**Usage**

```
dialogPlotOption_SkinCor(s)
```

**Arguments**

s                      integer number of Studies

**Value**

Dialog box with setting of correlation method

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/gbm_tcgaPlotTwoGenProf.rds", sep=""))
## Not run:
dialogPlotOption_SkinCor(1)

## End(Not run)
```

---

|                    |                                                                                                        |
|--------------------|--------------------------------------------------------------------------------------------------------|
| dialogSamplingGSEA | <i>Dialog Box for Sampling patients from expression profile data used for GSEA-R (Broad Institute)</i> |
|--------------------|--------------------------------------------------------------------------------------------------------|

---

**Description**

Dialog Box for Sampling patients from expression profile data used for GSEA-R (Broad Institute)

**Usage**

```
dialogSamplingGSEA(n_checked_GenProf, entryWidth = 10, returnValOnCancel = "ID_CANCEL")
```

**Arguments**

n\_checked\_GenProf  
                     Number of checked genetic profiles

entryWidth        10

returnValOnCancel  
                     "ID\_CANCEL"

**Value**

A vector with sampling size

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
Run.GSEA()
#dialogSamplingGSEA(1, entryWidth=10, returnValOnCancel = "ID_CANCEL")

## End(Not run)
```

---

```
dialogSelectFiles_GSEA
```

*Dialog Box to Select GCT, CLS, GMT and output Files for GSEA-R (Broad Institute)*

---

### Description

Dialog Box to Select GCT, CLS, GMT and output Files for GSEA-R (Broad Institute)

### Usage

```
dialogSelectFiles_GSEA()
```

### Value

A vector with files paths

### Examples

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
dialogSelectFiles_GSEA()

## End(Not run)
```

---

```
dialogSpecificMut
```

*dialog box to Specify Mutation using Regular Expression. Search specific mutation using regular expression.*

---

### Description

dialog box to Specify Mutation using Regular Expression. Search specific mutation using regular expression.

### Usage

```
getSpecificMut()
```

### Value

a dataframe with specific mutation informations

### Examples

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getSpecificMut()

## End(Not run)
```

---

|                    |                                                                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dialogSummary_GSEA | <i>Dialog Box to specify phenotype (variable) used in last GSEA-R to get Summary Results. This function ask the user to specify the phenotype (variable).</i> |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Description

Dialog Box to specify phenotype (variable) used in last GSEA-R to get Summary Results. This function ask the user to specify the phenotype (variable).

### Usage

```
dialogSummary_GSEA(Variable,returnValOnCancel ="ID_CANCEL")
```

### Arguments

|                   |             |
|-------------------|-------------|
| Variable          | phenotype   |
| returnValOnCancel | "ID_CANCEL" |

### Value

variables

### Examples

```
readRDS(paste(path.package("cancerR"),"/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
#Run.GSEA()
#getSummaryGSEA()

## End(Not run)
```

---

|                |                                      |
|----------------|--------------------------------------|
| displayInTable | <i>Display matrix in tcltk table</i> |
|----------------|--------------------------------------|

---

### Description

Display matrix in tcltk table

### Usage

```
displayInTable(tclarray,title="",height=-1,width=-1,nrow=-1,ncol=-1)
```

### Arguments

|          |                    |
|----------|--------------------|
| tclarray | a dataframe        |
| title    | title of the table |
| height   | -1                 |
| width    | -1                 |
| nrow     | -1                 |
| ncol     | -1                 |

**Value**

display a Table

**Examples**

```
#data(ClinicalData)
## Not run:
getInTable(Table= ClinicalData, title= "Clinical Data")

## End(Not run)
```

---

getCases

*Get cases for selected Studies. The Cases are the description of the samples from patients. The samples can be subdivided by the type of assays as, sequencing, CNA, Mutation, Methylation.*

---

**Description**

Get cases for selected Studies. The Cases are the description of the samples from patients. The samples can be subdivided by the type of assays as, sequencing, CNA, Mutation, Methylation.

**Usage**

```
getCases()
```

**Value**

a dataframe with cases

**Examples**

```
cgds <- cBioPortal(
  hostname = "www.cbioportal.org",
  protocol = "https",
  api = "/api/v2/api-docs"
)
## Not run:
getDataByGenes( api = cgds,
  studyId = "gbm_tcga_pub",
  genes = c("NF1", "TP53", "ABL1"),
  by = "hugoGeneSymbol",
  molecularProfileIds = "gbm_tcga_pub_mrna"
)

## End(Not run)
```

---

|                  |                                                            |
|------------------|------------------------------------------------------------|
| getCasesGenProfs | <i>get Cases and Genetic Profiles of selected Studies.</i> |
|------------------|------------------------------------------------------------|

---

**Description**

get Cases and Genetic Profiles of selected Studies.

**Usage**

```
getCasesGenProfs()
```

**Value**

This function is run by the "Get Cases and Genetic Profiles for selected Studies in starting window. This function needs to select at least one study and display Cases and genetic profiles in the main window.

**Examples**

```
cgds <- cBioPortal(
  hostname = "www.cbioportal.org",
  protocol = "https",
  api = "/api/v2/api-docs"
)
## Not run:
getDataByGenes( api = cgds,
  studyId = "gbm_tcga_pub",
  genes = c("NF1", "TP53", "ABL1"),
  by = "hugoGeneSymbol",
  molecularProfileIds = "gbm_tcga_pub_mrna"
)

## End(Not run)
```

---

|           |                                                                       |
|-----------|-----------------------------------------------------------------------|
| getCircos | <i>get Circos Layout for selected studies and selected dimensions</i> |
|-----------|-----------------------------------------------------------------------|

---

**Description**

get Circos Layout for selected studies and selected dimensions

**Usage**

```
getCircos(dimension)
```

**Arguments**

dimension      string (All,mRNA, CNA, Met,RPPA, miRNA, Mut)

**Value**

a plot with Circos style

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/Circos.rds", sep=""))
## Not run:
getCircos(dimension ="All")

## End(Not run)
```

---

```
getClinicalDataMatrix  get matrix with clinical from file
```

---

**Description**

get matrix with clinical from file

**Usage**

```
getClinicalDataMatrix()
```

**Value**

dataframe of clinicaldata

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
getClinicalDataMatrix()

## End(Not run)
```

---

```
getClinicData_MultipleCases
```

*get Clinical Data for Multiple Cases. User needs to select at least one case to run this function. Get clinical data for more one or multiple cases.*

---

**Description**

get Clinical Data for Multiple Cases. User needs to select at least one case to run this function. Get clinical data for more one or multiple cases.

**Usage**

```
getClinicData_MultipleCases(getSummaryGSEAExists)
```

**Arguments**

```
getSummaryGSEAExists
```

if equal to 0, the clinical data is displayed in table. if the argument is equal to 1, the clinical data is used to summarise GSEA analysis results.

**Value**

dataframe with clinical data

**Examples**

```
##Load Session
readRDS(paste(path.package("cancerR"),"/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Select Case
ENV <- new.env(parent = emptyenv())
ENV$curselectCases <- 2
## get Clinical data
## Not run:
getClinicData_MultipleCases(getSummaryGSEAExists = 0)

## End(Not run)
```

---

|                  |                                                      |
|------------------|------------------------------------------------------|
| getCor_ExpCNAMet | <i>Get gene correlation for multiple dimensions.</i> |
|------------------|------------------------------------------------------|

---

**Description**

Get gene correlation for multiple dimensions.

**Usage**

```
getCor_ExpCNAMet(ListMatrix, dimension)
```

**Arguments**

|            |                               |
|------------|-------------------------------|
| ListMatrix | is a List of numeric matrices |
| dimension  | Exp,CNA, Met , miRNA , RPPA   |

**Value**

correlation matrix

**Examples**

```
readRDS(paste(path.package("cancerR"),"/extdata/rdata/Circos.rds", sep=""))
## Not run:
getListProfData()
getCor_ExpCNAMet(ENV$ListProfData$Expression, dimension="mRNA")
head(ENV$Cor_Exp)

## End(Not run)
```

---

|         |                                                       |
|---------|-------------------------------------------------------|
| geteSet | <i>Built Expression Set (eSet) from profile data.</i> |
|---------|-------------------------------------------------------|

---

**Description**

Built Expression Set (eSet) from profile data.

**Usage**

```
geteSet()
```

**Value**

ExpressionSet

**Examples**

```
f <- 9
## Not run:
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))
geteSet()

## End(Not run)
```

---

|                |                               |
|----------------|-------------------------------|
| getFreqMutData | <i>get mutation frequency</i> |
|----------------|-------------------------------|

---

**Description**

get mutation frequency

**Usage**

```
getFreqMutData(list, GeneList)
```

**Arguments**

|          |                                                                           |
|----------|---------------------------------------------------------------------------|
| list     | a list of data frame with mutation data. Each data frame is for one study |
| GeneList | file name of GeneList examples: "73"                                      |

**Value**

a data frame with mutation frequency. gene is in rows and study is in column

**Examples**

```

cgds <- cBioPortal(
  hostname = "www.cbioportal.org",
  protocol = "https",
  api = "/api/v2/api-docs"
)
## Not run:
getDataByGenes( api = cgds,
  studyId = "gbm_tcga_pub",
  genes = c("NF1", "TP53", "ABL1"),
  by = "hugoGeneSymbol",
  molecularProfileIds = "gbm_tcga_pub_mrna"
)

## End(Not run)

```

---

|                  |                                       |
|------------------|---------------------------------------|
| getGCTCLSExample | <i>get GCT and CLS example files.</i> |
|------------------|---------------------------------------|

---

**Description**

get GCT and CLS example files.

**Usage**

```
getGCTCLSExample()
```

**Value**

GCT and CLS files

**Examples**

```

## Load workspace
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getGCTCLSExample()

## End(Not run)

```

---

|                 |                                                                           |
|-----------------|---------------------------------------------------------------------------|
| getGCT_CLSfiles | <i>get Profile (GCT file) and Phenotype (CLS file) Data from Disease.</i> |
|-----------------|---------------------------------------------------------------------------|

---

**Description**

get Profile (GCT file) and Phenotype (CLS file) Data from Disease.

**Usage**

```
getGCT_CLSfiles()
```

**Value**

GCT and CLS files paths

**Examples**

```
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getGCT_CLSfiles()

## End(Not run)
```

---

|                  |                                                  |
|------------------|--------------------------------------------------|
| getGeneExpMatrix | <i>get matrix with gene expression from file</i> |
|------------------|--------------------------------------------------|

---

**Description**

get matrix with gene expression from file

**Usage**

```
getGeneExpMatrix()
```

**Value**

dataframe of gene expression

**Examples**

```
readRDS(paste(path.package("cancer"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
getGeneExpMatrix()

## End(Not run)
```

---

|             |                                                                                                                                       |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|
| getGeneList | <i>User needs to specify which gene is interesting to get genomic cancer data. The gene must be with Symbol and one gene by line.</i> |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|

---

**Description**

User needs to specify which gene is interesting to get genomic cancer data. The gene must be with Symbol and one gene by line.

**Usage**

```
getGeneList()
```

**Value**

Gene list path of file

**Examples**

```
ENV <- new.env(parent = emptyenv())
## Not run:
getGeneList()

## End(Not run)
```

---

|                    |                                                                                  |
|--------------------|----------------------------------------------------------------------------------|
| getGeneListExample | <i>get Gene List from examples. User can select one from available gene list</i> |
|--------------------|----------------------------------------------------------------------------------|

---

**Description**

get Gene List from examples. User can select one from available gene list

**Usage**

```
getGeneListExample()
```

**Value**

Gene list path of file

**Examples**

```
ENV <- new.env(parent = emptyenv())
## Not run:
getGeneListExample()

## End(Not run)
```

---

|                       |                                  |
|-----------------------|----------------------------------|
| getGeneListFromMSigDB | <i>get gene list from MSigDB</i> |
|-----------------------|----------------------------------|

---

**Description**

get gene list from MSigDB

**Usage**

```
getGeneListFromMSigDB()
```

**Value**

a vector with gene list

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcgaGSEA1m1021.rds", sep=""))
## Not run:
getGeneListFromMSigDB()

## End(Not run)
```

---

|                    |                             |
|--------------------|-----------------------------|
| getGenesClassifier | <i>get Genes Classifier</i> |
|--------------------|-----------------------------|

---

**Description**

get Genes Classifier

**Usage**

```
getGenesClassifier()
```

**Value**

a data frma with genes classes

**Examples**

```
x <- 0
## Not run:
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
getGenesClassifier()

## End(Not run)
```

---

|                            |                                                                |
|----------------------------|----------------------------------------------------------------|
| getGenesTree_MultipleCases | <i>Get successively trees of genes list for multiple cases</i> |
|----------------------------|----------------------------------------------------------------|

---

**Description**

Get successively trees of genes list for multiple cases

**Usage**

```
getGenesTree_MultipleCases(entryWidth = 10)
```

**Arguments**

|            |    |
|------------|----|
| entryWidth | 10 |
|------------|----|

**Value**

plot tree

**Examples**

```
q <- readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
readRDS(paste(.libPaths(), "/cancerR/data/brca_tcga73genes.rds", sep=""))
getGenesTree_MultipleCases(entryWidth = 10)

## End(Not run)
```

---

`getGenesTree_SingleCase`*classify genes in tree for two phenotypes in the same case(disease).*

---

**Description**

classify genes in tree for two phenotypes in the same case(disease).

**Usage**

```
getGenesTree_SingleCase()
```

**Value**

tree plot

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))
## Not run:
getGenesTree_SingleCase()

## End(Not run)
```

---

`getGenProfs`*Get Genetic Profile from selected Studies*

---

**Description**

Get Genetic Profile from selected Studies

**Usage**

```
getGenProfs()
```

**Value**

dataframe with genetic profil

**Examples**

```
cgds <- cBioPortal(
  hostname = "www.cbioportal.org",
  protocol = "https",
  api = "/api/v2/api-docs"
)
## Not run:
getDataByGenes( api = cgds,
  studyId = "gbm_tcga_pub",
  genes = c("NF1", "TP53", "ABL1"),
  by = "hugoGeneSymbol",
```

```
molecularProfileIds = "gbm_tcga_pub_mrna"
)

## End(Not run)
```

---

|                    |                                                       |
|--------------------|-------------------------------------------------------|
| getGSEAlm_Diseases | <i>get GSEA linear modeling by studies (diseases)</i> |
|--------------------|-------------------------------------------------------|

---

### Description

get GSEA linear modeling by studies (diseases)

### Usage

```
getGSEAlm_Diseases()
```

### Value

a dataframe with annotation (GO, BP)

### Examples

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getGSEAlm_Diseases

## End(Not run)
```

---

|                     |                                                          |
|---------------------|----------------------------------------------------------|
| getGSEAlm_Variables | <i>get GSEA linear modeling by variables (phenotype)</i> |
|---------------------|----------------------------------------------------------|

---

### Description

get GSEA linear modeling by variables (phenotype)

### Usage

```
getGSEAlm_Variables()
```

### Value

a dataframe with annotation (GO, BP)

### Examples

```
x <- 3
## Not run:
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
getGSEAlm_Variables()

## End(Not run)
```

---

|            |                                      |
|------------|--------------------------------------|
| getInTable | <i>get dataframe in TK/TCL table</i> |
|------------|--------------------------------------|

---

**Description**

get dataframe in TK/TCL table

**Usage**

```
getInTable(table,title)
```

**Arguments**

|       |                             |
|-------|-----------------------------|
| table | Dataframe                   |
| title | string a title of the table |

**Value**

display a Table

**Examples**

```
#data(ClinicalData)
## Not run:
getInTable(Table= ClinicalData, title= "Clinical Data")

## End(Not run)
```

---

|                 |                                                                                       |
|-----------------|---------------------------------------------------------------------------------------|
| getListProfData | <i>Get list of data frame with profiles data (CNA,mRNA, Methylation, Mutation...)</i> |
|-----------------|---------------------------------------------------------------------------------------|

---

**Description**

Get list of data frame with profiles data (CNA,mRNA, Methylation, Mutation...)

**Usage**

```
getListProfData(checked_Studies, geneList)
```

**Arguments**

|                 |                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------|
| checked_Studies | checked studies in corresponding panel (input\$StudiesIDCircos, input\$StudiesIDReactome). |
| geneList        | GeneList with Hugo Symbol                                                                  |

**Value**

A LIST of profiles data (CNA, mRNA, Methylation, Mutation, miRNA, RPPA). Each dimension content a list of studies.

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
getListProfData()
head(ENV$ProfData$Expression)

## End(Not run)
```

---

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| getMegaProfData | <i>Get profile data for more than 500 genes list.</i> |
|-----------------|-------------------------------------------------------|

---

**Description**

Get profile data for more than 500 genes list.

**Usage**

```
getMegaProfData(MegaGeneList, k)
```

**Arguments**

|              |                           |
|--------------|---------------------------|
| MegaGeneList | Genelist >500             |
| k            | integer number of studies |

**Value**

dataframewith profile data

**Examples**

```
myGlobalEnv <- new.env(parent = emptyenv())
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcgaGSEA1m1021.rds", sep=""))
## Not run:
getMegaProfData(myGlobalEnv$MegaGeneList, 1)

## End(Not run)
```

---

|                         |                                                |
|-------------------------|------------------------------------------------|
| getMetDataMultipleGenes | <i>get Methylation data for multiple genes</i> |
|-------------------------|------------------------------------------------|

---

**Description**

get Methylation data for multiple genes

**Usage**

```
getMetDataMultipleGenes()
```

**Value**

a a dataframe with mean and median of methylation rate (threshold of silencing gene)

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getMetDataMultipleGenes()

## End(Not run)
```

---

getMSigDB

*Reduce MSigDB size for only gene list*


---

**Description**

Reduce MSigDB size for only gene list

**Usage**

```
getMSigDB(eSet, k)
```

**Arguments**

|      |                           |
|------|---------------------------|
| eSet | Expression Set            |
| k    | integer Number of studies |

**Value**

MSigDB for user gene List

**Examples**

```
d <- 7
## Not run:
setWorkspace()
getMSigDB(eSet = ENV$eSetClassifier, k = 1)

## End(Not run)
```

---

|                  |                                                               |
|------------------|---------------------------------------------------------------|
| getMSigDBExample | <i>get example of .gmt file from MSigDB (Broad Institute)</i> |
|------------------|---------------------------------------------------------------|

---

**Description**

get example of .gmt file from MSigDB (Broad Institute)

**Usage**

```
getMSigDBExample()
```

**Value**

path of GMT file

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
getMSigDBExample()

## End(Not run)
```

---

|               |                                                     |
|---------------|-----------------------------------------------------|
| getMSigDBfile | <i>Dialog Box to Select MSigDB Files from drive</i> |
|---------------|-----------------------------------------------------|

---

**Description**

Dialog Box to Select MSigDB Files from drive

**Usage**

```
getMSigDBfile()
```

**Value**

A path of MSigDB file

**Examples**

```
f <- 5+2
## Not run:
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))
geteSet()
getMSigDBfile()

## End(Not run)
```

---

|            |                                             |
|------------|---------------------------------------------|
| getMutData | <i>get Mutation data for multiple genes</i> |
|------------|---------------------------------------------|

---

**Description**

get Mutation data for multiple genes

**Usage**

```
getMutData()
```

**Value**

a dataframe with mutation informations

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getMutData()

## End(Not run)
```

---

|              |                                                 |
|--------------|-------------------------------------------------|
| getPhenoTest | <i>Associate phenotype to Studies (cancers)</i> |
|--------------|-------------------------------------------------|

---

**Description**

Associate phenotype to Studies (cancers)

**Usage**

```
getPhenoTest()
```

**Value**

a dataframe with disease/ variables association

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))
## Not run:
getPhenoTest(ENV$eSet)

## End(Not run)
```

---

|             |                                                                             |
|-------------|-----------------------------------------------------------------------------|
| getProfData | <i>Search and get genetic profiles (CNA,mRNA, Methylation, Mutation...)</i> |
|-------------|-----------------------------------------------------------------------------|

---

## Description

Search and get genetic profiles (CNA,mRNA, Methylation, Mutation...)

## Usage

```
getProfData(study,genProf, listGenProf, GeneList, Mut)
```

## Arguments

|             |                                                                            |
|-------------|----------------------------------------------------------------------------|
| study       | Study ID                                                                   |
| genProf     | Genetic Profile id (cancer_study_id_[mutations, cna, methylation, mrna ]). |
| listGenProf | A list of Genetic Profiles for one study.                                  |
| GeneList    | A list of genes                                                            |
| Mut         | Condition to set if the genetic profile is mutation or not (0,1)           |

## Details

See <https://github.com/kmezhound/bioCancer/wiki>

## Value

A data frame with Genetic profile

## Examples

```
cgds <- cBioPortal(
  hostname = "www.cbioportal.org",
  protocol = "https",
  api = "/api/v2/api-docs"
)
## Not run:
getDataByGenes( api = cgds,
  studyId = "gbm_tcga_pub",
  genes = c("NF1", "TP53", "ABL1"),
  by = "hugoGeneSymbol",
  molecularProfileIds = "gbm_tcga_pub_mrna"
)

## End(Not run)
```

---

```
getProfilesDataMultipleGenes
```

*get Profiles Data of multiple genes*

---

**Description**

get Profiles Data of multiple genes

**Usage**

```
getProfilesDataMultipleGenes(getSummaryGSEAEExists)
```

**Arguments**

getSummaryGSEAEExists  
if equal to 0, the clinical data is displayed in table. if the argument is equal to 1,  
the clinical data is used to summarise GSEA analysis results.

**Value**

a file with a dataframe of profile data

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))  
## Not run:  
getProfilesDataMultipleGenes(getSummaryGSEAEExists = 0)  
  
## End(Not run)
```

---

```
getProfilesDataSingleGene
```

*get Profiles Data for a Single Gene.*

---

**Description**

get Profiles Data for a Single Gene.

**Usage**

```
getProfilesDataSingleGene()
```

**Value**

dataframe with profiles data for a single gene

**Examples**

```

readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Select Case from Breast Cancer
ENV <- new.env(parent = emptyenv())
ENV$curselectCases <- 9
##Select Genetic Profile from Breast Cancer
ENV$curselectGenProfs <- 4
## get Specific Mutation data for 73 Genes list
## Not run:
getProfilesDataSingleGene()

## End(Not run)

```

---

|                |                                                      |
|----------------|------------------------------------------------------|
| getSpecificMut | <i>get specific Mutation data for multiple genes</i> |
|----------------|------------------------------------------------------|

---

**Description**

get specific Mutation data for multiple genes

**Usage**

```
getSpecificMut()
```

**Value**

a a dataframe with specific mutation informations

**Examples**

```

readRDS(paste(path.package("cancerR"), "/extdata/rdata//ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
getSpecificMut()

## End(Not run)

```

---

|                |                                                          |
|----------------|----------------------------------------------------------|
| getSummaryGSEA | <i>get Summary results from GSEA-R (Broad Institute)</i> |
|----------------|----------------------------------------------------------|

---

**Description**

get Summary results from GSEA-R (Broad Institute)

**Usage**

```
getSummaryGSEA()
```

**Value**

Dataframe with summary results

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
Run.GSEA()
getSummaryGSEA()

## End(Not run)
```

---

`getSurvival`*Survival plot*

---

**Description**

Survival plot

**Usage**

```
getSurvival(Coxph)
```

**Arguments**

|       |                                                                                            |
|-------|--------------------------------------------------------------------------------------------|
| Coxph | if Coxph = 0 : plot Kaplan-Meier curves else Coxph= 1 : plot Cox Proportional Hazard Model |
|-------|--------------------------------------------------------------------------------------------|

**Value**

Survival plot

**Examples**

```
surv <- 11
## Not run:
readRDS(paste(path.package("cancerR"), "/extdata/rdata/gbm_tcgaPlotTwoGenProf.rds", sep=""))
getSurvival(Coxph = 1)

## End(Not run)
```

---

`getTextWin`*get text in tcltk windows*

---

**Description**

get text in tcltk windows

**Usage**

```
getTextWin(text)
```

**Arguments**

|      |        |
|------|--------|
| text | string |
|------|--------|

**Value**

tcltk windows with text

**Examples**

```
text <- "mytext"
## Not run:
getTextWin(text)

## End(Not run)
```

---

|      |                                 |
|------|---------------------------------|
| GSEA | <i>GSEA-R (Broad Institute)</i> |
|------|---------------------------------|

---

**Description**

See [http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/R-GSEA\\_Readme](http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/R-GSEA_Readme)

**Value**

GSEA

**Author(s)**

Subramanian, Tamayo, et al. (2005, PNAS 102, 15545-15550) and Mootha, Lindgren, et al. (2003, Nat Genet 34, 267-273)

**Examples**

```
## Not run:
library(cancer)
## Load workspace
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
##Run.GSEA()

## End(Not run)
```

---

|                   |                          |
|-------------------|--------------------------|
| GSEA.Analyze.Sets | <i>GSEA.Analyze.Sets</i> |
|-------------------|--------------------------|

---

**Description**

[http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/R-GSEA\\_Readme](http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/R-GSEA_Readme)

**Usage**

```
GSEA.Analyze.Sets(directory, topgs="", non.interactive.run= FALSE, height=12, width=17)
```

**Arguments**

|                     |                            |
|---------------------|----------------------------|
| directory           | directory= fname.Output    |
| topgs               | topgs = 20                 |
| non.interactive.run | non.interactive.run= FALSE |
| height              | height=16                  |
| width               | width=16                   |

**Value**

GSEA.Analyze.Sets

**Author(s)**

Subramanian, Tamayo, et al. (2005, PNAS 102, 15545-15550) and Mootha, Lindgren, et al. (2003, Nat Genet 34, 267-273)

**References**

[http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/Main\\_Page](http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/Main_Page).

**Examples**

```
## Not run:
## Load workspace
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
##Run.GSEA()

## End(Not run)
```

---

GSEA.ConsPlot

*GSEA.ConsPlot*


---

**Description**

GSEA.ConsPlot

**Usage**

```
GSEA.ConsPlot(V, col.names, main = " ", sub = " ", xlab = " ", ylab = " ")
```

**Arguments**

|           |                      |
|-----------|----------------------|
| V         | V="Itable"           |
| col.names | col.names = colnames |
| main      | main= " "            |
| sub       | sub = " "            |
| xlab      | xlab= " "            |
| ylab      | ylab = " "           |

**Value**

GSEA.ConsPlot

**Examples**

```
## Not run:  
library(cancer)  
## Load workspace  
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))  
##Run.GSEA()  
  
## End(Not run)
```

---

GSEA.EnrichmentScore    *GSEA.EnrichmentScore*

---

**Description**

GSEA.EnrichmentScore

**Usage**

```
GSEA.EnrichmentScore(gene.list, gene.set, weighted.score.type = 1, correl.vector = NULL)
```

**Arguments**

gene.list  
gene.set  
weighted.score.type  
  
correl.vector

**Value**

GSEA.EnrichmentScore

**Examples**

```
## Not run:  
library(cancer)  
## Load workspace  
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))  
##Run.GSEA()  
  
## End(Not run)
```

---

|                       |                              |
|-----------------------|------------------------------|
| GSEA.EnrichmentScore2 | <i>GSEA.EnrichmentScore2</i> |
|-----------------------|------------------------------|

---

**Description**

GSEA.EnrichmentScore2

**Usage**

```
GSEA.EnrichmentScore2(gene.list, gene.set, weighted.score.type = 1, correl.vector = NULL)
```

**Arguments**

gene.list  
gene.set  
weighted.score.type

correl.vector

**Value**

GSEA.EnrichmentScore2

**Examples**

```
## Not run:  
library(cancer)  
## Load workspace  
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))  
##Run.GSEA()  
  
## End(Not run)
```

---

|                |                       |
|----------------|-----------------------|
| GSEA.Gct2Frame | <i>GSEA.Gct2Frame</i> |
|----------------|-----------------------|

---

**Description**

GSEA.Gct2Frame

**Usage**

```
GSEA.Gct2Frame(filename = "NULL")
```

**Arguments**

filename

**Value**

GSEA.GCT2Frame

Examples

```
## Not run:
library(canceR)
## Load workspace
readRDS(paste(path.package("canceR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
##Run.GSEA()

## End(Not run)
```

---

|                 |                        |
|-----------------|------------------------|
| GSEA.Gct2Frame2 | <i>GSEA.Gct2Frame2</i> |
|-----------------|------------------------|

---

Description

GSEA.Gct2Frame2

Usage

```
GSEA.Gct2Frame2(filename = "NULL")
```

Arguments

filename

Value

GSEA.GCT2Frame2

Examples

```
## Not run:
library(canceR)
## Load workspace
readRDS(paste(path.package("canceR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
##Run.GSEA()

## End(Not run)
```

---

|                  |                         |
|------------------|-------------------------|
| GSEA.GeneRanking | <i>GSEA.GeneRanking</i> |
|------------------|-------------------------|

---

Description

GSEA.GeneRanking

**Arguments**

A  
class.labels  
gene.labels  
nperm  
permutation.type  
  
sigma.correction  
  
fraction  
replace  
reverse.sign

**Value**

GSEA.GeneRanking

**Examples**

```
## Not run:  
library(cancerR)  
## Load workspace  
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))  
##Run.GSEA()  
  
## End(Not run)
```

---

GSEA.HeatMapPlot

*GSEA.HeatMapPlot*

---

**Description**

GSEA.HeatMapPlot

**Value**

GSEA.HeatMapPlot

**Examples**

```
## Not run:  
library(cancerR)  
## Load workspace  
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))  
##Run.GSEA()  
  
## End(Not run)
```

---

|                   |                          |
|-------------------|--------------------------|
| GSEA.HeatMapPlot2 | <i>GSEA.HeatMapPlot2</i> |
|-------------------|--------------------------|

---

**Description**

GSEA.HeatMapPlot2

**Value**

GSEA.HeatMapPlot2

**Examples**

```
## Not run:
library(cancer)
## Load workspace
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
##Run.GSEA()

## End(Not run)
```

---

|                    |                           |
|--------------------|---------------------------|
| GSEA.NormalizeCols | <i>GSEA.NormalizeCols</i> |
|--------------------|---------------------------|

---

**Description**

GSEA.NormalizeCols

**Usage**

GSEA.NormalizeCols(V)

**Arguments**

V

**Value**

GSEA.NormalizeCols

**Examples**

```
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
## Load workspace
##Run.GSEA()

## End(Not run)
```

---

|                    |                           |
|--------------------|---------------------------|
| GSEA.NormalizeRows | <i>GSEA.NormalizeRows</i> |
|--------------------|---------------------------|

---

**Description**

GSEA.NormalizeRows

**Usage**

GSEA.NormalizeRows(V)

**Arguments**

V

**Value**

GSEA.NormalizeRows

**Examples**

```
## Not run:
library(cancer)
## Load workspace
readRDS(paste(path.package("cancer"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
##Run.GSEA()

## End(Not run)
```

---

|                  |                         |
|------------------|-------------------------|
| GSEA.ReadClsFile | <i>GSEA.ReadClsFile</i> |
|------------------|-------------------------|

---

**Description**

GSEA.ReadClsFile

**Usage**

GSEA.ReadClsFile(file = "NULL")

**Arguments**

file

**Value**

GSEA.ReadClsFile

Examples

```
readRDS(paste(path.package("cancerR"),"/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
##Run.GSEA()

## End(Not run)
```

---

|                |                       |
|----------------|-----------------------|
| GSEA.Res2Frame | <i>GSEA.Res2Frame</i> |
|----------------|-----------------------|

---

Description

GSEA.Res2Frame

Usage

```
GSEA.Res2Frame(filename = "NULL")
```

Arguments

filename

Value

GSEA.NormalizeCols

Examples

```
readRDS(paste(path.package("cancerR"),"/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
##Run.GSEA()

## End(Not run)
```

---

|                |                       |
|----------------|-----------------------|
| GSEA.Threshold | <i>GSEA.Threshold</i> |
|----------------|-----------------------|

---

Description

GSEA.Threshold

Usage

```
GSEA.Threshold(V, thres, ceil)
```

Arguments

V  
thres  
ceil

**Value**

GSEA.Threshold

**Examples**

```
## Load workspace
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:

##Run.GSEA()

## End(Not run)
```

---

GSEA.VarFilter

*GSEA.VarFilter*

---

**Description**

GSEA.VarFilter

**Usage**

```
GSEA.VarFilter(V, fold, delta, gene.names = "NULL")
```

**Arguments**

V  
fold  
delta  
gene.names

**Value**

GSEA.VarFilter

**Examples**

```
## Load workspace
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
##Run.GSEA()

## End(Not run)
```

---

|                |                       |
|----------------|-----------------------|
| GSEA.write.gct | <i>GSEA.write.gct</i> |
|----------------|-----------------------|

---

**Description**

GSEA.write.gct

**Usage**

GSEA.write.gct(gct, filename)

**Arguments**

gct  
filename

**Value**

GSEA.Write.gct

**Examples**

```
## Load workspace
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:

##Run.GSEA()

## End(Not run)
```

---

Match\_GeneList\_MSigDB    *Search MSigDb that overlap gene list*

---

**Description**

Search MSigDb that overlap gene list

**Usage**

Match\_GeneList\_MSigDB

**Value**

GeneList

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/prad_michPhenoTest1021.rds", sep=""))
## Not run:
Match_GeneList_MSigDB()

## End(Not run)
```

---

|             |                                           |
|-------------|-------------------------------------------|
| modalDialog | <i>Dialog box to specify Gene Symbol.</i> |
|-------------|-------------------------------------------|

---

**Description**

Dialog box to specify Gene Symbol.

**Usage**

```
modalDialog(title, question, entryInit, entryWidth = 40, returnValOnCancel = "ID_CANCEL")
```

**Arguments**

|                   |             |
|-------------------|-------------|
| title             | string      |
| question          | string      |
| entryInit         | entryInit   |
| entryWidth        | 40          |
| returnValOnCancel | "ID_CANCEL" |

**Value**

dialog box

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Select Case from Breast Cancer
ENV <- new.env(parent = emptyenv())
ENV$curselectCases <- 9
##Select Genetic Profile from Breast Cancer
ENV$curselectGenProfs <- 4
## get Specific Mutation data for 73 Genes list
## Not run:
getProfilesDataSingleGene()

## End(Not run)
```

---

|             |                    |
|-------------|--------------------|
| myGlobalEnv | <i>myGlobalEnv</i> |
|-------------|--------------------|

---

**Description**

Global environment to store cancer variables.

**Format**

The format is: <environment: 0xb3eb240>

**Examples**

```
myGlobalEnv <- new.env(parent = emptyenv())
```

---

OLD.GSEA.EnrichmentScore

*OLD.GSEA.EnrichmentScore*

---

**Description**

OLD.GSEA.EnrichmentScore

**Arguments**

gene.list  
gene.set

**Value**

OLD.GSEA.EnchmentScore

**Examples**

```
readRDS(paste(path.package("cancer"),"/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
##Run.GSEA()

## End(Not run)
```

---

plotModel

*model plotting with tcltk*

---

**Description**

model plotting with tcltk

**Usage**

```
plotModel(plotCommand, title= "TITLE",hscale=1, vscale=1 )
```

**Arguments**

|             |                  |
|-------------|------------------|
| plotCommand | plotcommand      |
| title       | title of plot    |
| hscale      | horizintal scale |
| vscale      | vertical scale   |

**Value**

plot

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/gbm_tcgaPlotTwoGenProf.rds", sep=""))
## Not run:
plot_1Gene_2GenProfs()

## End(Not run)
```

---

plot\_1Gene\_2GenProfs    *Plotting two genetic profiles for one Gene*

---

**Description**

Plotting two genetic profiles for one Gene

**Usage**

```
plot_1Gene_2GenProfs()
```

**Value**

plot

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/gbm_tcgaPlotTwoGenProf.rds", sep=""))
## Not run:
plot_1Gene_2GenProfs()

## End(Not run)
```

---

plot\_2Genes\_1GenProf    *plot correlation of two genes expressions.*

---

**Description**

plot correlation of two genes expressions.

**Usage**

```
plot_2Genes_1GenProf()
```

**Value**

plot

**Examples**

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
plot_2Genes_1GenProf()

## End(Not run)
```

---

|                       |                           |
|-----------------------|---------------------------|
| <code>rbind.na</code> | <i>bind non equal row</i> |
|-----------------------|---------------------------|

---

**Description**

bind non equal row

**Usage**

```
rbind.na(..., deparse.level = 1)
```

**Arguments**

`...` (generalized) vectors or matrices.

`deparse.level` integer controlling the construction of labels in the case of non-matrix-like arguments (for the default method): `deparse.level = 0` constructs no labels; the default, `deparse.level = 1` or `2` constructs labels from the argument names.

**Value**

a data frame with merged rows

**Examples**

```
## Not run:
row1 <- c("a", "b", "c", "d")
row2 <- c("A", "B", "C")
row3 <- rbind.na(row1, row2)

## End(Not run)
```

---

|                       |                                                             |
|-----------------------|-------------------------------------------------------------|
| <code>Run.GSEA</code> | <i>The main function to run GSEA-R from Broad Institute</i> |
|-----------------------|-------------------------------------------------------------|

---

**Description**

The main function to run GSEA-R from Broad Institute

**Usage**

```
Run.GSEA()
```

**Value**

A vector with sampling size

Examples

```
readRDS(paste(path.package("cancerR"),"/extdata/rdata/ucec_tcga_pubGSEA1021.rds", sep=""))
## Not run:
Run.GSEA()

## End(Not run)
```

---

|              |                                                                                                                                                           |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| setWorkspace | <i>Setting work Directory and output folders.At starting window, user needs to set work directory for output data. The function is foud in File menu.</i> |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|

---

Description

Setting work Directory and output folders.At starting window, user needs to set work directory for output data. The function is foud in File menu.

Usage

```
setWorkspace()
```

Value

paths of output files

Examples

```
readRDS(paste(path.package("cancerR"),"/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
setWorkspace()

## End(Not run)
```

---

|           |                                                |
|-----------|------------------------------------------------|
| test.CGDS | <i>S3 method to test cBioPortal connection</i> |
|-----------|------------------------------------------------|

---

Description

S3 method to test cBioPortal connection

Usage

```
## S3 method for class 'CGDS'
test(x, ...)
```

Arguments

|     |                   |
|-----|-------------------|
| x   | connection object |
| ... | not used          |

---

```
testCheckedCaseGenProf
```

*Testing checked appropriate Cases for appropriate Genetic profiles.*

---

### Description

Testing checked appropriate Cases for appropriate Genetic profiles.

### Usage

```
testCheckedCaseGenProf(singleGene=0)
```

### Arguments

singleGene      specify if the check for querying genetic profile for a specific gene or not (0,1).

### Value

dialog box with warning message

### Examples

```
readRDS(paste(path.package("cancerR"), "/extdata/rdata/brca_tcga73genes.rds", sep=""))
## Not run:
testCheckedCaseGenProf(singleGene=0)

## End(Not run)
```

---

```
UnifyRowNames
```

*Unify row names in data frame with the same order of gene list.*

---

### Description

Unify row names in data frame with the same order of gene list.

### Usage

```
UnifyRowNames(x, geneList)
```

### Arguments

x                      data frame with gene symbol in the row name  
geneList              a gene list

### Value

a data frame having the gene in row name ordered as in gene list.

**Examples**

```
cgds <- cBioPortal(  
  hostname = "www.cbioportal.org",  
  protocol = "https",  
  api = "/api/v2/api-docs"  
)  
## Not run:  
getDataByGenes( api = cgds,  
  studyId = "gbm_tcga_pub",  
  genes = c("NF1", "TP53", "ABL1"),  
  by = "hugoGeneSymbol",  
  molecularProfileIds = "gbm_tcga_pub_mrna"  
)  
  
## End(Not run)
```

# Index

## \* datasets

myGlobalEnv, 45

about, 3

cancerR, 4

cancerR\_Issue, 4

cancerR\_Vignette, 5

cbind.na, 5

CGDS, 6

dialogGeneClassifier, 6

dialoggetGeneListMSigDB, 7

dialogMetOption, 8

dialogMut, 8

dialogOptionCircos, 9

dialogOptionGSEAlm, 9

dialogOptionPhenoTest, 10

dialogPlotOption\_SkinCor, 10

dialogSamplingGSEA, 11

dialogSelectFiles\_GSEA, 12

dialogSpecificMut, 12

dialogSummary\_GSEA, 13

displayInTable, 13

getCases, 14

getCasesGenProfs, 15

getCircos, 15

getClinicalDataMatrix, 16

getClinicData\_MultipleCases, 16

getCor\_ExpCNAMet, 17

geteSet, 18

getFreqMutData, 18

getGCT\_CLSfiles, 19

getGCTCLSExample, 19

getGeneExpMatrix, 20

getGeneList, 20

getGeneListExample, 21

getGeneListFromMSigDB, 21

getGenesClassifier, 22

getGenesTree\_MultipleCases, 22

getGenesTree\_SingleCase, 23

getGenProfs, 23

getGSEAlm\_Diseases, 24

getGSEAlm\_Variables, 24

getInTable, 25

getListProfData, 25

getMegaProfData, 26

getMetDataMultipleGenes, 26

getMSigDB, 27

getMSigDBExample, 28

getMSigDBfile, 28

getMutData, 29

getPhenoTest, 29

getProfData, 30

getProfilesDataMultipleGenes, 31

getProfilesDataSingleGene, 31

getSpecificMut, 32

getSummaryGSEA, 32

getSurvival, 33

getTextWin, 33

GSEA, 34

GSEA.Analyze.Sets, 34

GSEA.ConsPlot, 35

GSEA.EnrichmentScore, 36

GSEA.EnrichmentScore2, 37

GSEA.Gct2Frame, 37

GSEA.Gct2Frame2, 38

GSEA.GeneRanking, 38

GSEA.HeatMapPlot, 39

GSEA.HeatMapPlot2, 40

GSEA.NormalizeCols, 40

GSEA.NormalizeRows, 41

GSEA.ReadClsFile, 41

GSEA.Res2Frame, 42

GSEA.Threshold, 42

GSEA.VarFilter, 43

GSEA.write.gct, 44

Match\_GeneList\_MSigDB, 44

modalDialog, 45

myGlobalEnv, 45

OLD.GSEA.EnrichmentScore, 46

plot\_1Gene\_2GenProfs, 47

plot\_2Genes\_1GenProf, 47

plotModel, 46

`rbind.na`, [48](#)

`Run.GSEA`, [48](#)

`setWorkspace`, [49](#)

`test.CGDS`, [49](#)

`testCheckedCaseGenProf`, [50](#)

`UnifyRowNames`, [50](#)