

# Package ‘GDCRNATools’

November 21, 2024

**Title** GDCRNATools: an R/Bioconductor package for integrative analysis of lncRNA, mRNA, and miRNA data in GDC

**Version** 1.26.0

**Author** Ruidong Li,  
Han Qu,  
Shibo Wang,  
Julong Wei,  
Le Zhang,  
Renyuan Ma,  
Jianming Lu,  
Jianguo Zhu,  
Wei-De Zhong,  
Zhenyu Jia

**Maintainer** Ruidong Li <rli012@ucr.edu>,  
Han Qu <hqu002@ucr.edu>

**Description** This is an easy-to-use package for downloading, organizing, and integrative analyzing RNA expression data in GDC with an emphasis on deciphering the lncRNA-mRNA related ceRNA regulatory network in cancer. Three databases of lncRNA-miRNA interactions including spongeScan, starBase, and miRcode, as well as three databases of mRNA-miRNA interactions including miRTarBase, starBase, and miRcode are incorporated into the package for ceRNAs network construction. limma, edgeR, and DESeq2 can be used to identify differentially expressed genes/miRNAs. Functional enrichment analyses including GO, KEGG, and DO can be performed based on the clusterProfiler and DO packages. Both univariate CoxPH and KM survival analyses of multiple genes can be implemented in the package. Besides some routine visualization functions such as volcano plot, bar plot, and KM plot, a few simply shiny apps are developed to facilitate visualization of results on a local webpage.

**Depends** R (>= 3.5.0)

**License** Artistic-2.0

**Encoding** UTF-8

**LazyData** false

**Imports** shiny, jsonlite, rjson, XML, limma, edgeR, DESeq2, clusterProfiler, DOSE, org.Hs.eg.db, biomaRt, survival, survminer, pathview, ggplot2, gplots, DT, GenomicDataCommons, BiocParallel

**Suggests** knitr, testthat, prettydoc, rmarkdown

**VignetteBuilder** knitr

**biocViews** ImmunoOncology, GeneExpression, DifferentialExpression, GeneRegulation, GeneTarget, NetworkInference, Survival, Visualization, GeneSetEnrichment, NetworkEnrichment, Network, RNASeq, GO, KEGG

**RoxygenNote** 6.1.0

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** d8c163d

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-11-20

## Contents

GDCRNATools-package	3
DEGAll	3
enrichOutput	3
gdcBarPlot	3
gdcCEAnalysis	4
gdcClinicalDownload	6
gdcClinicalMerge	7
gdcCorPlot	8
gdcDEAnalysis	9
gdcDEReport	10
gdcEnrichAnalysis	11
gdcEnrichPlot	12
gdcExportNetwork	13
gdcFilterDuplicate	14
gdcFilterSampleType	14
gdcHeatmap	15
gdcKMPlot	16
gdcMatchSamples	17
gdcParseMetadata	18
gdcRNADownload	18
gdcRNAMerge	19
gdcSurvivalAnalysis	20
gdcVolcanoPlot	21
gdcVoomNormalization	22
lncTarget	23
mirCounts	23
pcTarget	23
rnaCounts	23

<i>GDCRNATools-package</i>	3
shinyCorPlot . . . . .	24
shinyKMPlot . . . . .	25
shinyPathview . . . . .	26

<b>Index</b>	<b>27</b>
--------------	-----------

---

GDCRNATools-package	<i>This is an easy-to-use package for downloading, organizing, and integrative analyzing RNA expression data in GDC with an emphasis on deciphering the lncRNA-mRNA related ceRNA regulatory network in cancer.</i>
---------------------	---

---

**Description**

This is an easy-to-use package for downloading, organizing, and integrative analyzing RNA expression data in GDC with an emphasis on deciphering the lncRNA-mRNA related ceRNA regulatory network in cancer.

---

DEGAll	<i>Output of <a href="#">gdcDEAnalysis</a> for downstream analysis</i>
--------	--

---

**Description**

Output of [gdcDEAnalysis](#) for downstream analysis

---

enrichOutput	<i>Output of <a href="#">gdcEnrichAnalysis</a> for visualization</i>
--------------	--

---

**Description**

Output of [gdcEnrichAnalysis](#) for visualization

---

gdcBarPlot	<i>Bar plot of differentially expressed genes/miRNAs</i>
------------	--

---

**Description**

A bar plot showing the number of down-regulated and up-regulated DE genes/miRNAs of different biotypes

**Usage**

```
gdcBarPlot(deg, angle = 0, data.type)
```

**Arguments**

deg	a dataframe generated from <a href="#">gdcDEReport</a> containing DE genes/miRNAs ids, logFC, etc.
angle	a numeric value specifying the angle of text on x-axis. Default is 0
data.type	one of 'RNAseq' and 'miRNAs'

**Value**

A bar plot

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
genes <- c('ENSG00000231806', 'ENSG00000261211', 'ENSG00000260920',
           'ENSG00000228594', 'ENSG00000125170', 'ENSG00000179909',
           'ENSG00000280012', 'ENSG00000134612', 'ENSG00000213071')
symbol <- c('PCAT7', 'AL031123.2', 'AL031985.3',
           'FNDC10', 'DOK4', 'ZNF154',
           'RPL23AP61', 'FOLH1B', 'LPAL2')
group <- rep(c('long_non_coding', 'protein_coding', 'pseudogene'), each=3)
logFC <- c(2.8, 2.3, -1.1, 1.9, -1.2, -1.6, 1.5, 2.1, -1.1)
FDR <- rep(c(0.1, 0.00001, 0.0002), each=3)
deg <- data.frame(symbol, group, logFC, FDR)
rownames(deg) <- genes
gdcBarPlot(deg, angle=45, data.type='RNAseq')
```

---

gdcCEAnalysis

*Competing endogenous RNAs (ceRNAs) analysis*

---

**Description**

Identify ceRNAs by (1) number of shared miRNAs between lncRNA and mRNA; (2) expression correlation of lncRNA and mRNA; (3) regulation similarity of shared miRNAs on lncRNA and mRNA; (4) sensitivity correlation

**Usage**

```
gdcCEAnalysis(lnc, pc, deMIR = NULL, lnc.targets = "starBase",
             pc.targets = "starBase", rna.expr, mir.expr)
```

**Arguments**

lnc	a vector of Ensembl long non-coding gene ids
pc	a vector of Ensembl protein coding gene ids
deMIR	a vector of differentially expressed miRNAs. Default is NULL
lnc.targets	a character string specifying the database of miRNA-lncRNA interactions. Should be one of 'spongeScan', 'starBase', and 'miRcode'. Default is 'starBase'.

Or a list of miRNA-lncRNA interactions generated by users



```
ceOutput <- gdcCEAnalysis(lnc      = deLNC,
                          pc       = dePC,
                          lnc.targets = 'starBase',
                          pc.targets = 'starBase',
                          rna.expr  = rnaExpr,
                          mir.expr  = mirExpr)
```

---

gdcClinicalDownload    *Download clinical data in GDC*

---

## Description

Download clinical data in GDC either by providing the manifest file or specifying the project id and data type

## Usage

```
gdcClinicalDownload(manifest = NULL, project.id,
                    directory = "Clinical", write.manifest = FALSE,
                    method = "gdc-client")
```

## Arguments

manifest	manifest file that is downloaded from the GDC cart. If provided, files whose UUIDs are in the manifest file will be downloaded via gdc-client, otherwise, project id argument should be provided to download data automatically. Default is NULL
project.id	project id in GDC
directory	the folder to save downloaded files. Default is 'Clinical'
write.manifest	logical, whether to write out the manifest file
method	method that is used to download data. Either 'GenomicDataCommons' which is a well established method developed in the <b>GenomicDataCommons</b> package, or alternatively 'gdc-client' which uses the gdc-client tool developed by GDC. Default is 'gdc-client'.

## Value

downloaded files in the specified directory

## Author(s)

Ruidong Li and Han Qu

## Examples

```
##### Download Clinical data by manifest file #####
manifest <- 'Clinical.manifest.txt'
## Not run: gdcClinicalDownload(manifest = manifest,
                              directory = 'Clinical')
## End(Not run)

##### Download Clinical data by project id #####
```

```
project <- 'TCGA-PRAD'  
## Not run: gdcClinicalDownload(project.id = project,  
                               write.manifest = TRUE,  
                               directory = 'Clinical')  
## End(Not run)
```

---

gdcClinicalMerge      *Merge clinical data*

---

## Description

Merge clinical data in .xml files that are downloaded from GDC to a dataframe

## Usage

```
gdcClinicalMerge(path, key.info = TRUE, organized = FALSE)
```

## Arguments

path	path to downloaded files for merging
key.info	logical, whether to return the key clinical information only. If TRUE, only clinical information such as age, stage, grade, overall survival, etc. will be returned
organized	logical, whether the clinical data have already been organized into a single folder (eg., data downloaded by the 'GenomicDataCommons' method are already organized). Default is FALSE.

## Value

A dataframe of clinical data with rows are patients and columns are clinical traits

## Author(s)

Ruidong Li and Han Qu

## Examples

```
##### Merge clinical data #####  
path <- 'Clinical/'  
## Not run: clinicalDa <- gdcClinicalMerge(path=path, key.info=TRUE)
```

gdcCorPlot

*Correlation plot of two genes/miRNAs***Description**

Scatter plot showing the expression correlation between two genes/miRNAs

**Usage**

```
gdcCorPlot(gene1, gene2, rna.expr, metadata)
```

**Arguments**

gene1	an Ensembl gene id or miRBase v21 mature miRNA id
gene2	an Ensembl gene id or miRBase v21 mature miRNA id
rna.expr	<a href="#">voom</a> transformed expression data
metadata	metadata parsed from <a href="#">gdcParseMetadata</a>

**Value**

A scatter plot with line of best fit

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
genes <- c('ENSG00000000938', 'ENSG00000000971', 'ENSG00000001036',
           'ENSG00000001084', 'ENSG00000001167', 'ENSG00000001460')

samples <- c('TCGA-2F-A9K0-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-11',
            'TCGA-2F-A9KT-11', 'TCGA-2F-A9KW-11')

metaMatrix <- data.frame(sample_type=rep(c('PrimaryTumor',
                                           'SolidTissueNormal'),each=3),
                         sample=samples,
                         days_to_death=seq(100,600,100),
                         days_to_last_follow_up=rep(NA,6))

rnaExpr <- matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,
                   0.5,2.5,5.7,6.5,4.9,3.8,
                   2.1,2.9,5.9,5.7,4.5,3.5,
                   2.7,5.9,4.5,5.8,5.2,3.0,
                   2.5,2.2,5.3,4.4,4.4,2.9,
                   2.4,3.8,6.2,3.8,3.8,4.2),6,6)

rownames(rnaExpr) <- genes
colnames(rnaExpr) <- samples
gdcCorPlot(gene1 = 'ENSG00000000938',
           gene2 = 'ENSG00000001084',
           rna.expr = rnaExpr,
           metadata = metaMatrix)
```



---

`gdcDEAnalysis`*Differential gene expression analysis*

---

**Description**

Performs differential gene expression analysis by **limma**, **edgeR**, and **DESeq2**

**Usage**

```
gdcDEAnalysis(counts, group, comparison, method = "limma",  
              n.cores = NULL, filter = TRUE)
```

**Arguments**

<code>counts</code>	a dataframe or numeric matrix of raw counts data generated from <a href="#">gdcRNAMerge</a>
<code>group</code>	a vector giving the group that each sample belongs to
<code>comparison</code>	a character string specifying the two groups being compared. Example: <code>comparison='PrimaryTumor-SolidTissueNormal'</code>
<code>method</code>	one of 'limma', 'edgeR', and 'DESeq2'. Default is 'limma' Note: It may takes long time for <code>method='DESeq2'</code> with a single core
<code>n.cores</code>	a numeric value of cores to be used for <code>method='DESeq2'</code> to accelate the analysis process. Default is NULL
<code>filter</code>	logical, whether to filter out low expression genes. If TRUE, only genes with <code>cpm &gt; 1</code> in more than half of the samples will be kept. Default is TRUE

**Value**

A dataframe containing Ensembl gene ids/miRBase v21 mature miRNA ids, gene symbols, biotypes, fold change on the log2 scale, p value, and FDR etc. of all genes/miRNAs of analysis.

**Note**

It may takes long time for `method='DESeq2'` with a single core. Please use multiple cores if possible

**Author(s)**

Ruidong Li and Han Qu

**References**

Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2010 Jan 1;26(1):139-40.  
Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth GK. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research*. 2015 Jan 20; 43(7):e47-e47.  
Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology*. 2014 Dec 5; 15(12):550.

**Examples**

```
genes <- c('ENSG0000000938', 'ENSG0000000971', 'ENSG0000001036',
           'ENSG0000001084', 'ENSG0000001167', 'ENSG0000001460')

samples <- c('TCGA-2F-A9K0-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-11',
            'TCGA-2F-A9KT-11', 'TCGA-2F-A9KW-11')

metaMatrix <- data.frame(sample_type=rep(c('PrimaryTumor',
                                           'SolidTissueNormal'), each=3),
                        sample=samples,
                        days_to_death=seq(100, 600, 100),
                        days_to_last_follow_up=rep(NA, 6))
rnaMatrix <- matrix(c(6092, 11652, 5426, 4383, 3334, 2656,
                    8436, 2547, 7943, 3741, 6302, 13976,
                    1506, 6467, 5324, 3651, 1566, 2780,
                    834, 4623, 10275, 5639, 6183, 4548,
                    24702, 43, 1987, 269, 3322, 2410,
                    2815, 2089, 3804, 230, 883, 5415), 6, 6)
rownames(rnaMatrix) <- genes
colnames(rnaMatrix) <- samples
DEGall <- gdcDEAnalysis(counts = rnaMatrix,
                       group = metaMatrix$sample_type,
                       comparison = 'PrimaryTumor-SolidTissueNormal',
                       method = 'limma')
```

gdcDEReport

*Report differentially expressed genes/miRNAs***Description**

Report genes/miRNAs that are differentially expressed satisfying a given threshold

**Usage**

```
gdcDEReport(deg, gene.type = "all", fc = 2, pval = 0.01)
```

**Arguments**

deg	A dataframe of DE analysis result from <a href="#">gdcDEAnalysis</a>
gene.type	one of 'all', 'long_non_coding', 'protein_coding', and 'miRNAs'. Default is 'all'
fc	a numeric value specifying the threshold of fold change
pval	a numeric value specifying the threshold of p value

**Value**

A dataframe or numeric matrix of differentially expressed genes/miRNAs

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
genes <- c('ENSG0000000938', 'ENSG0000000971', 'ENSG0000001036',
           'ENSG0000001084', 'ENSG0000001167', 'ENSG0000001460')

samples <- c('TCGA-2F-A9KO-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-11',
            'TCGA-2F-A9KT-11', 'TCGA-2F-A9KW-11')

metaMatrix <- data.frame(sample_type=rep(c('PrimaryTumor',
                                           'SolidTissueNormal'), each=3),
                        sample=samples,
                        days_to_death=seq(100, 600, 100),
                        days_to_last_follow_up=rep(NA, 6))

rnaMatrix <- matrix(c(6092, 11652, 5426, 4383, 3334, 2656,
                    8436, 2547, 7943, 3741, 6302, 13976,
                    1506, 6467, 5324, 3651, 1566, 2780,
                    834, 4623, 10275, 5639, 6183, 4548,
                    24702, 43, 1987, 269, 3322, 2410,
                    2815, 2089, 3804, 230, 883, 5415), 6, 6)

rownames(rnaMatrix) <- genes
colnames(rnaMatrix) <- samples
DEGAll <- gdcDEAnalysis(counts = rnaMatrix,
                       group = metaMatrix$sample_type,
                       comparison = 'PrimaryTumor-SolidTissueNormal',
                       method = 'limma')

dePC <- gdcDEReport(deg=DEGAll)
```

---

gdcEnrichAnalysis      *Functional enrichment analysis*

---

**Description**

Performs Gene Ontology (GO), Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway and Disease Ontology (DO) enrichment analyses by **clusterProfiler** and **DOSE** packages

**Usage**

```
gdcEnrichAnalysis(gene, simplify = TRUE, level = 0)
```

**Arguments**

gene	a vector of Ensembl gene id
simplify	logical, specifying whether to remove redundant GO terms. Default simplify=TRUE
level	a numeric value, restrict the GO enrichment result at a specific GO level. Default is 0, which means all terms should be returned

**Value**

A dataframe of enrichment analysis result containing enriched terms, number of overlapped genes, p value of hypergeometric test, fdr, fold of enrichment, Ensembl gene ids, gene symbols, and functional categories, etc.

**Author(s)**

Ruidong Li and Han Qu

**References**

Yu G, Wang LG, Han Y, He QY. clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*. 2012 May 1;16(5):284-7.

Yu G, Wang LG, Yan GR, He QY. DOSE: an R/Bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics*. 2014 Oct 17;31(4):608-9.

**Examples**

```
##### GO, KEGG, DO enrichment analysis #####
deg <- c('ENSG0000000938', 'ENSG0000000971', 'ENSG0000001036',
        'ENSG0000001084', 'ENSG0000001167', 'ENSG0000001460')
## Not run: enrichOutput <- gdcEnrichAnalysis(gene=deg, simplify=TRUE)
```

---

gdcEnrichPlot

*Plots for enrichment analysis*

---

**Description**

Bar plot and bubble plot for GO, KEGG, and DO functional enrichment analysis

**Usage**

```
gdcEnrichPlot(enrichment, type = "bar", category = "KEGG",
              num.terms = 10, bar.color = "black")
```

**Arguments**

enrichment	a dataframe generated from <a href="#">gdcEnrichAnalysis</a>
type	type of the plot, should be one of 'bar' and 'bubble'
category	which category should be plotted. Possible values are 'KEGG', 'GO', 'GO_BP', 'GO_CC', 'GO_MF', and 'DO'. Default is 'KEGG'
num.terms	number of terms to be plotted. Default is 10
bar.color	color of the bar plot. Default is 'black'

**Value**

A bar plot or bubble plot of functional enrichment analysis

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### Enrichment plots #####
enrichOutput<-data.frame(Terms=c('hsa05414~Dilated cardiomyopathy (DCM)',
                                'hsa04510~Focal adhesion',
                                'hsa05205~Proteoglycans in cancer'),
                        Category=rep('KEGG',3),
                        FDR=c(0.001,0.002,0.003))
gdcEnrichPlot(enrichment=enrichOutput, type='bar', category='KEGG')
```

---

gdcExportNetwork	<i>Export network for Cytoscape</i>
------------------	-------------------------------------

---

**Description**

Export nodes and edges of ce network for **Cytoscape** visualization

**Usage**

```
gdcExportNetwork(ceNetwork, net)
```

**Arguments**

ceNetwork	a dataframe generated from <a href="#">gdcCEAnalysis</a>
net	one of 'nodes' and 'edges'

**Value**

A dataframe of nodes or edges

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### ceRNA network analysis #####
ceOutput <- data.frame(lncRNAs=c('ENSG00000242125', 'ENSG00000242125',
                                'ENSG00000245532'),
                      Genes=c('ENSG0000043355', 'ENSG00000109586',
                               'ENSG00000144355'),
                      miRNAs=c('hsa-miR-340-5p', 'hsa-miR-340-5p',
                                'hsa-miR-320b, hsa-miR-320d,
                                hsa-miR-320c, hsa-miR-320a'),
                      Counts=c(1,1,4), stringsAsFactors=FALSE)
##### Export edges #####
edges <- gdcExportNetwork(ceNetwork=ceOutput, net='edges')

##### Export nodes #####
## Not run: nodes <- gdcExportNetwork(ceNetwork=ceOutput, net='nodes')
```

gdcFilterDuplicate     *Filter out duplicated samples*

---

**Description**

Filter out samples that are sequenced for two or more times

**Usage**

```
gdcFilterDuplicate(metadata)
```

**Arguments**

metadata            metadata parsed from [gdcParseMetadata](#)

**Value**

A filtered dataframe of metadata without duplicated samples

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### Parse metadata by project id and data type #####  
metaMatrix <- gdcParseMetadata(project.id='TARGET-RT', data.type='RNAseq')  
metaMatrix <- gdcFilterDuplicate(metadata=metaMatrix)
```

---

gdcFilterSampleType     *Filter out other type of samples*

---

**Description**

Filter out samples that are neither *Solid Tissue Normal* nor *Primary Tumor*

**Usage**

```
gdcFilterSampleType(metadata)
```

**Arguments**

metadata            metadata parsed from [gdcParseMetadata](#)

**Value**

A filtered dataframe of metadata with *Solid Tissue Normal* and *Primary Tumor* samples only

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### Parse metadata by project id and data type #####
metaMatrix <- gdcParseMetadata(project.id='TARGET-RT', data.type='RNAseq')
metaMatrix <- gdcFilterSampleType(metadata=metaMatrix)
```

gdcHeatmap

*Heatmap of differentially expressed genes/miRNAs***Description**

A heatmap showing unsupervised hierarchical clustering of DE genes/miRNAs by [heatmap.2](#) in the **gplots** package

**Usage**

```
gdcHeatmap(deg.id, metadata, rna.expr)
```

**Arguments**

deg.id	a vector of Ensembl gene ids or miRBase v21 mature miRNA ids
metadata	metadata parsed from <a href="#">gdcParseMetadata</a>
rna.expr	<a href="#">voom</a> transformed expression data

**Value**

A heatmap with rows are DE genes/miRNAs and columns are samples. *Solid Tissue Normal* samples are labeled with blue and *Primary Tumor* samples are labeled with red

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
genes <- c('ENSG00000000938', 'ENSG00000000971', 'ENSG00000001036',
           'ENSG00000001084', 'ENSG00000001167', 'ENSG00000001460')

samples <- c('TCGA-2F-A9K0-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-01',
            'TCGA-2F-A9KT-01', 'TCGA-2F-A9KW-01')

metaMatrix <- data.frame(sample_type=rep('PrimaryTumor',6),
                        sample=samples,
                        days_to_death=seq(100,600,100),
                        days_to_last_follow_up=rep(NA,6))

rnaExpr <- matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,
                  0.5,2.5,5.7,6.5,4.9,3.8,
                  2.1,2.9,5.9,5.7,4.5,3.5,
                  2.7,5.9,4.5,5.8,5.2,3.0,
                  2.5,2.2,5.3,4.4,4.4,2.9,
                  2.4,3.8,6.2,3.8,3.8,4.2),6,6)

rownames(rnaExpr) <- genes
colnames(rnaExpr) <- samples
gdcHeatmap(deg.id=genes, metadata=metaMatrix, rna.expr=rnaExpr)
```

gdcKMPlot

*Kaplan Meier plot***Description**

Plot Kaplan Meier survival curve

**Usage**

```
gdcKMPlot(gene, rna.expr, metadata, sep = "median")
```

**Arguments**

gene	an Ensembl gene id
rna.expr	<a href="#">voom</a> transformed expression data
metadata	metadata parsed from <a href="#">gdcParseMetadata</a>
sep	a character string specifying which point should be used to separate low-expression and high-expression groups. Possible values are '1stQu', 'mean', 'median', and '3rdQu'. Default is 'median'

**Value**

A plot of Kaplan Meier survival curve

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### KM plots #####
genes <- c('ENSG00000000938', 'ENSG00000000971', 'ENSG00000001036',
          'ENSG00000001084', 'ENSG00000001167', 'ENSG00000001460')

samples <- c('TCGA-2F-A9K0-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-01',
            'TCGA-2F-A9KT-01', 'TCGA-2F-A9KW-01')

metaMatrix <- data.frame(sample_type=rep('PrimaryTumor',6),
                        sample=samples,
                        days_to_death=seq(100,600,100),
                        days_to_last_follow_up=rep(NA,6))
rnaExpr <- matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,
                  0.5,2.5,5.7,6.5,4.9,3.8,
                  2.1,2.9,5.9,5.7,4.5,3.5,
                  2.7,5.9,4.5,5.8,5.2,3.0,
                  2.5,2.2,5.3,4.4,4.4,2.9,
                  2.4,3.8,6.2,3.8,3.8,4.2),6,6)
rownames(rnaExpr) <- genes
colnames(rnaExpr) <- samples
gdcKMPlot(gene='ENSG00000000938', rna.expr=rnaExpr,
          metadata=metaMatrix, sep='median')
```



---

gdcMatchSamples	<i>Match samples in metadata and expression matrix</i>
-----------------	--

---

## Description

Check if samples in the metadata and expression data match

## Usage

```
gdcMatchSamples(metadata, rna.expr)
```

## Arguments

metadata	metadata parsed from <a href="#">gdcParseMetadata</a>
rna.expr	<a href="#">voom</a> transformed expression data

## Value

A logical value. If TRUE, all the samples matched

## Author(s)

Ruidong Li and Han Qu

## Examples

```
genes <- c('ENSG00000000938', 'ENSG00000000971', 'ENSG00000001036',
           'ENSG00000001084', 'ENSG00000001167', 'ENSG00000001460')

samples <- c('TCGA-2F-A9KO-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-01',
            'TCGA-2F-A9KT-01', 'TCGA-2F-A9KW-01')

metaMatrix <- data.frame(sample_type=rep('PrimaryTumor',6),
                        sample=samples,
                        days_to_death=seq(100,600,100),
                        days_to_last_follow_up=rep(NA,6))
rnaExpr <- matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,
                  0.5,2.5,5.7,6.5,4.9,3.8,
                  2.1,2.9,5.9,5.7,4.5,3.5,
                  2.7,5.9,4.5,5.8,5.2,3.0,
                  2.5,2.2,5.3,4.4,4.4,2.9,
                  2.4,3.8,6.2,3.8,3.8,4.2),6,6)

rownames(rnaExpr) <- genes
colnames(rnaExpr) <- samples
gdcMatchSamples(metadata=metaMatrix, rna.expr=rnaExpr)
```

---

gdcParseMetadata      *Parse metadata*

---

### Description

Parse metadata either by providing the *.json* file that is downloaded from GDC cart or by parse metadata automatically by providing the project id and data type

### Usage

```
gdcParseMetadata(metafile = NULL, project.id, data.type,
  write.meta = FALSE)
```

### Arguments

metafile	metadata file in <i>.json</i> format download from GDC cart. If provided, the meta-data will be parsed from this file, otherwise, project and data.type arguments should be provided to retrieve metadata automatically. Default is NULL
project.id	project id in GDC
data.type	one of 'RNAseq' and 'miRNAs'
write.meta	logical, whether to write the metadata to a <i>.json</i> file

### Value

A dataframe of metadata containing file\_name, sample\_id, etc. as well as some basic clinical data

### Author(s)

Ruidong Li and Han Qu

### Examples

```
##### Merge RNA expression data #####
metaMatrix <- gdcParseMetadata(project.id='TARGET-RT', data.type='RNAseq')
```

---

gdcRNADownload      *Download RNA data in GDC*

---

### Description

Download gene expression quantification and isoform expression quantification data from GDC either by providing the manifest file or by sepcifying the project id and data type

### Usage

```
gdcRNADownload(manifest = NULL, project.id, data.type,
  directory = "Data", write.manifest = FALSE, method = "gdc-client")
```

**Arguments**

manifest	manifest file that is downloaded from the GDC cart. If provided, files whose UUIDs are in the manifest file will be downloaded via gdc-client, otherwise, project and data.type arguments should be provided to download data automatically. Default is NULL
project.id	project id in GDC
data.type	one of 'RNAseq' and 'miRNAs'
directory	the folder to save downloaded files. Default is 'Data'
write.manifest	logical, whether to write out the manifest file
method	method that is used to download data. Either 'GenomicDataCommons' which is a well established method developed in the <b>GenomicDataCommons</b> package, or alternatively 'gdc-client' which uses the gdc-client tool developed by GDC. Default is 'gdc-client'.

**Value**

Downloaded files in the specified directory

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### Download RNA data by manifest file #####
manifest <- 'RNAseq.manifest.txt'
## Not run: gdcRNADownload(manifest=manifest)

##### Download RNA data by project id and data type #####
project <- 'TCGA-PRAD'
## Not run: gdcRNADownload(project.id=project, data.type='RNAseq')
```

---

gdcRNAMerge

---

*Merge RNA/miRNAs raw counts data*


---

**Description**

Merge raw counts data that is downloaded from GDC to a single expression matrix

**Usage**

```
gdcRNAMerge(metadata, path, data.type, organized = FALSE)
```

**Arguments**

metadata	metadata parsed from <a href="#">gdcParseMetadata</a>
path	path to downloaded files for merging
data.type	one of 'RNAseq' and 'miRNAs'
organized	logical, whether the raw counts data have already been organized into a single folder (eg., data downloaded by the 'GenomicDataCommons' method are already organized). Default is FALSE.

**Value**

A dataframe or numeric matrix of raw counts data with rows are genes or miRNAs and columns are samples

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
##### Merge RNA expression data #####
metaMatrix <- gdcParseMetadata(project.id='TARGET-RT',
  data.type='RNAseq')
## Not run: rnaExpr <- gdcRNAMerge(metadata=metaMatrix, path='RNAseq/',
  data.type='RNAseq')
## End(Not run)
```

---

gdcSurvivalAnalysis    *Univariate survival analysis of multiple genes*

---

**Description**

Univariate Cox Proportional-Hazards and Kaplan Meier survival analysis of a vector of genes

**Usage**

```
gdcSurvivalAnalysis(gene, rna.expr, metadata, method = "coxph",
  sep = "median")
```

**Arguments**

gene	a vector of Ensembl gene ids
rna.expr	<a href="#">voom</a> transformed expression data
metadata	metadata parsed from <a href="#">gdcParseMetadata</a>
method	method for survival analysis. Possible values are 'coxph' and 'KM'. Default is 'coxph'
sep	which point should be used to separate low-expression and high-expression groups for method='KM'. Possible values are '1stQu', 'mean', 'median', and '3rdQu'. Default is 'median'

**Value**

A dataframe or numeric matrix of hazard ratio, 95% confidence interval, p value, and FDR

**Author(s)**

Ruidong Li and Han Qu

## References

- Therneau TM, Lumley T. Package ‘survival’.
- Andersen PK, Gill RD. Cox’s regression model for counting processes: a large sample study. *The annals of statistics*. 1982 Dec 1;1100-20.
- Therneau TM, Grambsch PM. Extending the Cox model. Edited by P. Bickel, P. Diggle, S. Fienberg, K. Krickeberg. 2000:51.
- Harrington DP, Fleming TR. A class of rank test procedures for censored survival data. *Biometrika*. 1982 Dec 1;69(3):553-66.

## Examples

```
genes <- c('ENSG00000000938', 'ENSG00000000971', 'ENSG00000001036',
          'ENSG00000001084', 'ENSG00000001167', 'ENSG00000001460')

samples <- c('TCGA-2F-A9K0-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-01',
            'TCGA-2F-A9KT-01', 'TCGA-2F-A9KW-01')

metaMatrix <- data.frame(sample_type=rep('PrimaryTumor',6),
                        sample=samples,
                        days_to_death=seq(100,600,100),
                        days_to_last_follow_up=rep(NA,6))

rnaExpr <- matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,
                  0.5,2.5,5.7,6.5,4.9,3.8,
                  2.1,2.9,5.9,5.7,4.5,3.5,
                  2.7,5.9,4.5,5.8,5.2,3.0,
                  2.5,2.2,5.3,4.4,4.4,2.9,
                  2.4,3.8,6.2,3.8,3.8,4.2),6,6)

rownames(rnaExpr) <- genes
colnames(rnaExpr) <- samples
survOutput <- gdcSurvivalAnalysis(gene=genes,
                                rna.expr=rnaExpr, metadata=metaMatrix)
```

---

gdcVolcanoPlot

*Volcano plot of differentially expressed genes/miRNAs*

---

## Description

A volcano plot showing differentially expressed genes/miRNAs

## Usage

```
gdcVolcanoPlot(deg.all, fc = 2, pval = 0.01)
```

## Arguments

- |         |  |
|---------|--|
| deg.all | a dataframe generated from <a href="#">gdcDEAnalysis</a> containing all genes of analysis no matter they are differentially expressed or not |
| fc      | a numeric value specifying the threshold of fold change  |
| pval    | a numeric value specifying the threshold of p value  |

**Value**

A volcano plot

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
genes <- c('ENSG00000231806', 'ENSG00000261211', 'ENSG00000260920',
           'ENSG00000228594', 'ENSG00000125170', 'ENSG00000179909',
           'ENSG00000280012', 'ENSG00000134612', 'ENSG00000213071')
symbol <- c('PCAT7', 'AL031123.2', 'AL031985.3',
            'FNDC10', 'DOK4', 'ZNF154',
            'RPL23AP61', 'FOLH1B', 'LPAL2')
group <- rep(c('long_non_coding', 'protein_coding', 'pseudogene'), each=3)
logFC <- c(2.8, 2.3, -1.1, 1.9, -1.2, -1.6, 1.5, 2.1, -1.1)
FDR <- rep(c(0.1, 0.00001, 0.0002), each=3)
deg <- data.frame(symbol, group, logFC, FDR)
rownames(deg) <- genes
gdcVolcanoPlot(deg.all=deg)
```

---

gdcVoomNormalization *TMM normalization and voom transformation*

---

**Description**

Normalize raw counts data by TMM implemented in **edgeR** and then transform it by **voom** in **limma**

**Usage**

```
gdcVoomNormalization(counts, filter = TRUE)
```

**Arguments**

counts	raw counts of RNA/miRNA expression data
filter	logical, whether to filter out low-expression genes. If TRUE, only genes with cpm > 1 in more than half of the samples will be kept. Default is TRUE

**Value**

A dataframe or numeric matrix of TMM normalized and **voom** transformed expression values on the log<sub>2</sub> scale

**Author(s)**

Ruidong Li and Han Qu

**References**

Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2010 Jan 1;26(1):139-40.  
 Law CW, Chen Y, Shi W, Smyth GK. Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome biology*. 2014 Feb 3;15(2):R29.

**Examples**

```
##### Normalization #####
rnaMatrix <- matrix(sample(1:100,100), 4, 25)
rnaExpr <- gdcVoomNormalization(counts=rnaMatrix, filter=FALSE)
```

---

IncTarget	<i>miRNA-lncRNA interactions</i>
-----------	----------------------------------

---

**Description**

miRNA-lncRNA interactions

---

mirCounts	<i>miRNA counts data of TCGA-CHOL</i>
-----------	---------------------------------------

---

**Description**

miRNA counts data of TCGA-CHOL

---

pcTarget	<i>miRNA-mRNA interactions</i>
----------	--------------------------------

---

**Description**

miRNA-mRNA interactions

---

rnaCounts	<i>RNAseq counts data of TCGA-CHOL</i>
-----------	--

---

**Description**

RNAseq counts data of TCGA-CHOL





---

`shinyKMPlot`*Shiny Kaplan Meier (KM) plot*

---

## Description

A simple **shiny** app to show KM survival curves on local web browser

## Usage

```
shinyKMPlot(gene, rna.expr, metadata)
```

## Arguments

<code>gene</code>	a vector of Ensembl gene ids
<code>rna.expr</code>	<a href="#">voom</a> transformed expression data
<code>metadata</code>	metadata parsed from <a href="#">gdcParseMetadata</a>

## Value

a local webpage for visualization of KM plots

## Author(s)

Ruidong Li and Han Qu

## Examples

```
genes <- c('ENSG0000000938', 'ENSG0000000971', 'ENSG0000001036',
           'ENSG0000001084', 'ENSG0000001167', 'ENSG0000001460')

samples <- c('TCGA-2F-A9K0-01', 'TCGA-2F-A9KP-01',
            'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-01',
            'TCGA-2F-A9KT-01', 'TCGA-2F-A9KW-01')

metaMatrix <- data.frame(sample_type=rep('PrimaryTumor',6),
                        sample=samples,
                        days_to_death=seq(100,600,100),
                        days_to_last_follow_up=rep(NA,6))

rnaExpr <- matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,
                  0.5,2.5,5.7,6.5,4.9,3.8,
                  2.1,2.9,5.9,5.7,4.5,3.5,
                  2.7,5.9,4.5,5.8,5.2,3.0,
                  2.5,2.2,5.3,4.4,4.4,2.9,
                  2.4,3.8,6.2,3.8,3.8,4.2),6,6)

rownames(rnaExpr) <- genes
colnames(rnaExpr) <- samples
## Not run: shinyKMPlot(gene=genes, rna.expr=rnaExpr,
  metadata=metaMatrix)
## End(Not run)
```

---

shinyPathview	<i>Shiny pathview</i>
---------------	-----------------------

---

**Description**

A simple **shiny** app to show pathways generated by **pathview** package on local web browser

**Usage**

```
shinyPathview(gene, pathways, directory = ".")
```

**Arguments**

gene	a vector of numeric values (eg. fold change on log2 scale) with names are Ensembl gene ids
pathways	a vector of KEGG pathway ids
directory	the folder to save pathway figures. Default is the working directory

**Value**

a local webpage for visualization of KEGG maps

**Author(s)**

Ruidong Li and Han Qu

**Examples**

```
genes <- c('ENSG0000000938', 'ENSG0000000971', 'ENSG0000001036',
           'ENSG0000001084', 'ENSG0000001167', 'ENSG0000001460')
pathways <- c("hsa05414~Dilated cardiomyopathy (DCM)",
             "hsa05410~Hypertrophic cardiomyopathy (HCM)",
             "hsa05412~Arrhythmogenic right ventricular cardiomyopathy",
             "hsa04512~ECM-receptor interaction",
             "hsa04510~Focal adhesion",
             "hsa04360~Axon guidance",
             "hsa04270~Vascular smooth muscle contraction",
             "hsa05205~Proteoglycans in cancer",
             "hsa04022~cGMP-PKG signaling pathway",
             "hsa00480~Glutathione metabolism")
## Not run: shinyPathview(gene=genes, pathways=pathways)
```

# Index

## \* datasets

- DEGAll, 3
- enrichOutput, 3
- lncTarget, 23
- mirCounts, 23
- pcTarget, 23
- rnaCounts, 23

rnaCounts, 23

- shinyCorPlot, 24
- shinyKMPlot, 25
- shinyPathview, 26

voom, 5, 8, 15–17, 20, 22, 24, 25

DEGAll, 3

enrichOutput, 3

- gdcBarPlot, 3
- gdcCEAnalysis, 4, 13
- gdcClinicalDownload, 6
- gdcClinicalMerge, 7
- gdcCorPlot, 8
- gdcDEAnalysis, 3, 9, 10, 21
- gdcDEReport, 4, 10
- gdcEnrichAnalysis, 3, 11, 12
- gdcEnrichPlot, 12
- gdcExportNetwork, 13
- gdcFilterDuplicate, 14
- gdcFilterSampleType, 14
- gdcHeatmap, 15
- gdcKMPlot, 16
- gdcMatchSamples, 17
- gdcParseMetadata, 8, 14–17, 18, 19, 20, 24, 25
- gdcRNADownload, 18
- gdcRNAMerge, 9, 19
- GDCRNATools (GDCRNATools-package), 3
- GDCRNATools-package, 3
- gdcSurvivalAnalysis, 20
- gdcVolcanoPlot, 21
- gdcVoomNormalization, 22

heatmap.2, 15

lncTarget, 23

mirCounts, 23

pcTarget, 23