GDCRNATools: integrative analysis of protein coding genes, long non-coding genes, and microRNAs in GDC

Ruidong Li and Han Qu

Last update: 29 November, 2017

1 Introduction

GDCRNATools is an R package which provides a standard, easy-to-use and comprehensive pipeline for downloading, organizing, and integrative analyzing RNA expression data in the GDC portal with an emphasis on deciphering the lncRNA-mRNA related ceRNA regulatory network in cancer.

Competing endogenous RNAs (ceRNAs) are RNAs that indirectly regulate other transcripts by competing for shared miRNAs. Although only a fraction of long non-coding RNAs has been functionally characterized, increasing evidences show that lncRNAs harboring multiple miRNA response elements (MREs) can act as ceRNAs to sequester miRNA activity and thus reduce the inhibition of miRNA on its targets. Deregulation of ceRNAs network may lead to human diseases.

The Genomic Data Commons (GDC) maintains standardized genomic, clinical, and biospecimen data from National Cancer Institute (NCI) programs including The Cancer Genome Atlas (TCGA) and Therapeutically Applicable Research To Generate Effective Treatments (TARGET), It also accepts high quality datasets from non-NCI supported cancer research programs, such as genomic data from the Foundation Medicine.

Many analyses can be perfomed using GDCRNATools, including differential gene expression analysis (limmaref, edgeRref, and DESeq2ref), univariate survival analysis (CoxPH and KM), competing endogenous RNA network analysis (hypergeometric test, Pearson correlation analysis, regulation similarity analysis, sensitivity Pearson partial correlationref), and functional enrichment analysis(GO, KEGG, DO). Besides some routine visualization methods such as volcano plot, scatter plot, and bubble plot, etc., three simple shiny apps are developed in GDCRNATools allowing users visualize the results on a local webpage. All the figures are plotted based on ggplot2 package unless otherwise specified.

This user-friendly package allows researchers perform the analysis by simply running a few functions and integrate their own pipelines such as molecular subtype classification, weighted correlation network analysis (WGCNA)ref, and TF-miRNA co-regulatory network analysis, etc. into the workflow easily. This could open a door to accelerate the study of crosstalk among different classes of RNAs and their regulatory relationships in cancer.

2 `GDCRNATools` package installation

The R software for running `GDCRNATools` can be downloaded from The Comprehensive R Archive Network (CRAN). The `GDCRNATools` package can be installed from Github.

```
devtools::install_github(repo='Jialab-UCR/GDCRNATools')
```

```
library(GDCRNATools)
```

3 Data download

Two methods are provided for downloading Gene Expression Quantification (HTSeq-Counts), Isoform Expression Quantification (BCGSC miRNA Profiling), and Clinical (Clinical Supplement) data:

- Manual download
  Step1: Download GDC Data Transfer Tool on the GDC website
  Step2: Add data to the GDC cart, then download manifest file and metadata of the cart

Step3: Download data using `gdcRNADownload()` function by providing the manifest file

- Automatic download
  Download GDC Data Transfer Tool, manifest file, and data automatically by specifying the `project.id` and `data.type` in `gdcRNADownload()` function for RNAseq and miRNAs data, and in `gdcClinicalDownload()` function for clinical data

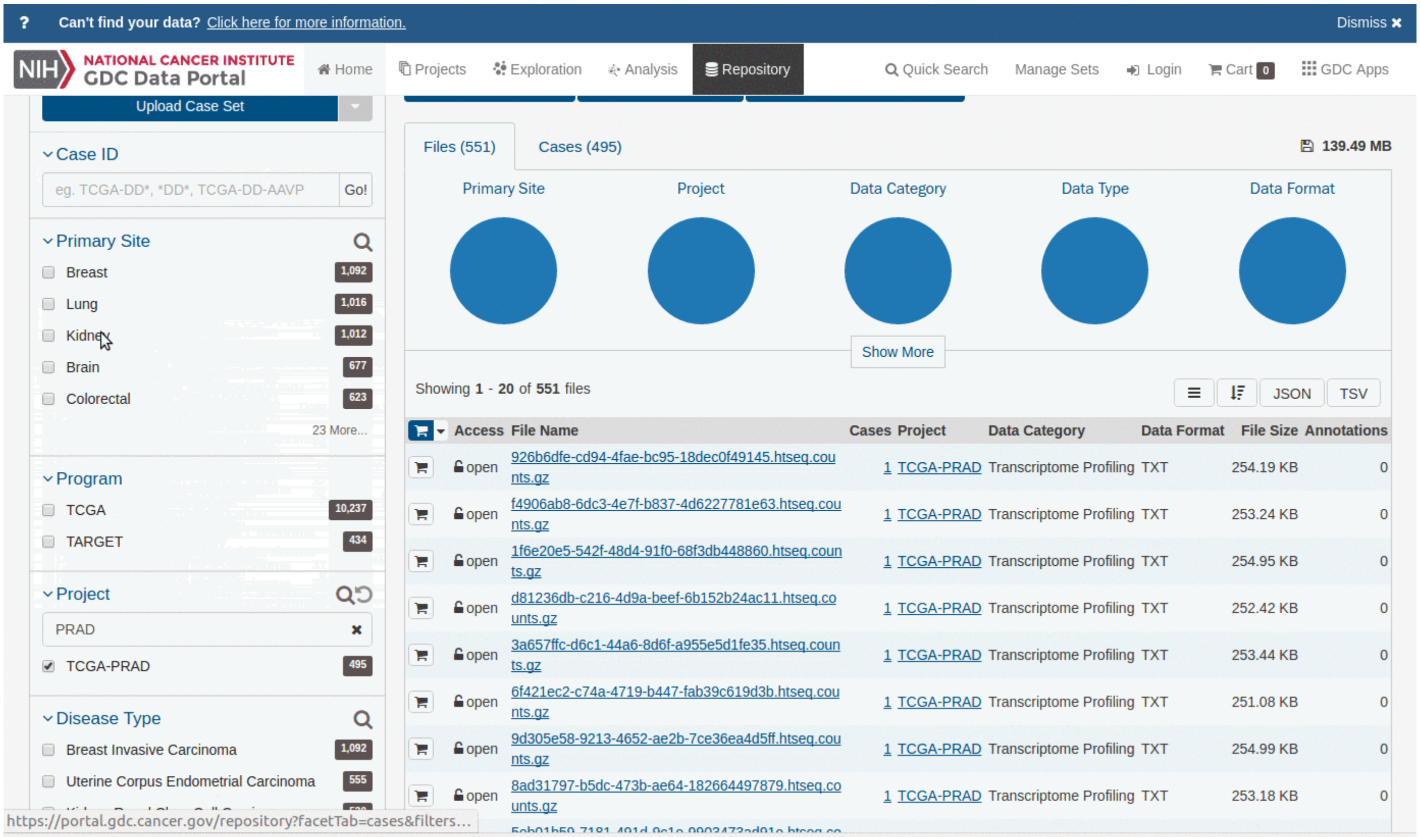
Users can also download data from GDC using the API method developed in TCGAbiolinksref or using TCGA-Assemblerref

3.1 Manual download

3.1.1 Installation of GDC Data Transfer Tool gdc-client

Download GDC Data Transfer Tool from the GDC website and unzip the file

3.1.2 Download manifest file and metadata from GDC Data Portal



Case ID

eg. TCGA-DD*, *DD*, TCGA-DD-AAVP Go!

Primary Site

□ Breast	1,092
□ Lung	1,016
□ Kidney	1,012
□ Brain	677
□ Colorectal	623

23 More...

Program

□ TCGA	10,237
□ TARGET	434

Project

PRAD ✖

☑ TCGA-PRAD	495

Disease Type

□ Breast Invasive Carcinoma	1,092
□ Uterine Corpus Endometrial Carcinoma	555

Files (551) Cases (495) 139.49 MB

Showing **1** - **20** of **551** files

	Access	File Name	Cases	Project	Data Category	Data Format	File Size	Annotations
	open	926b6dfe-cd94-4fae-bc95-18dec0f49145.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	254.19 KB	0
	open	f4906ab8-6dc3-4e7f-b837-4d6227781e63.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	253.24 KB	0
	open	1f6e20e5-542f-48d4-91f0-68f3db448860.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	254.95 KB	0
	open	d81236db-c216-4d9a-beef-6b152b24ac11.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	252.42 KB	0
	open	3a657ffc-d6c1-44a6-8d6f-a955e5d1fe35.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	253.44 KB	0
	open	6f421ec2-c74a-4719-b447-fab39c619d3b.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	251.08 KB	0
	open	9d305e58-9213-4652-ae2b-7ce36ea4d5ff.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	254.99 KB	0
	open	8ad31797-b5dc-473b-ae64-182664497879.htseq.counts.gz	1	TCGA-PRAD	Transcriptome Profiling	TXT	253.18 KB	0

https://portal.gdc.cancer.gov/repository?facetTab=cases&filters...

3.1.3 Download data

```
####### Download RNAseq data #######
gdcRNADownload(manifest  = 'TCGA-PRAD/TCGA-PRAD.RNAseq.gdc_manifest.2017
               directory = 'TCGA-PRAD/RNAseq')

####### Download miRNAs data #######
gdcRNADownload(manifest  = 'TCGA-PRAD/TCGA-PRAD.miRNAs.gdc_manifest.2017
               directory = 'TCGA-PRAD/miRNAs')

####### Download Clinical data #######
gdcRNADownload(manifest  = 'TCGA-PRAD/TCGA-PRAD.Clinical.gdc_manifest.20
               directory = 'TCGA-PRAD/Clinical')
```

3.2 Automatic download

- `gdcRNADownload()` will download HTSeq-Counts data if `data.type='RNAseq'` and download BCGSC miRNA Profiling data if `data.type='miRNAs'`. `project.id` argument is required to be provided.
- `gdcClinicalDownload()` download clinical data in .xml format automatically by simply specifying the `project.id` argument.

3.2.1 Download RNAseq/miRNAs data

```
####### Download RNAseq data #######
gdcRNADownload(project.id     = 'TCGA-PRAD',
               data.type      = 'RNAseq',
               write.manifest = TRUE,
               directory      = 'TCGA-PRAD/RNAseq')

####### Download miRNAs data #######
gdcRNADownload(project.id     = 'TCGA-PRAD',
               data.type      = 'miRNAs',
               write.manifest = TRUE,
               directory      = 'TCGA-PRAD/miRNAs')
```

3.2.2 Download clinical data

```
####### Download clinical data #######
gdcClinicalDownload(project.id     = 'TCGA-PRAD',
                    data.type      = 'RNAseq',
                    write.manifest = TRUE,
                    directory      = 'TCGA-PRAD/Clinical')
```

4 Data organization

4.1 Parse metadata

Metadata can be parsed by either providing the metadata file that is downloaded in the data download step, or specifying the `project.id` and `data.type` in `gdcParseMetadata()` function to obtain information of data in the manifest file to facilitate data organization and basic clinical information of patients such as age, stage and gender, etc. for data analysis.

4.1.1 Parse metadata by providing the metadata file

```
####### Parse RNAseq metadata #######
metaMatrix.RNA <- gdcParseMetadata(metafile='TCGA-PRAD/TCGA-PRAD.RNAseq.me

####### Parse miRNAs metadata #######
metaMatrix.MIR <- gdcParseMetadata(metafile='TCGA-PRAD/TCGA-PRAD.miRNAs.me
```

4.1.2 Parse metadata by specifying project.id and data.type

```
####### Parse RNAseq metadata #######
metaMatrix.RNA <- gdcParseMetadata(project.id = 'TCGA-PRAD',
                                   data.type  = 'RNAseq',
                                   write.meta = TRUE)

####### Parse miRNAs metadata #######
metaMatrix.MIR <- gdcParseMetadata(project.id = 'TCGA-PRAD',
                                   data.type  = 'miRNAs',
                                   write.meta = TRUE)
metaMatrix.MIR[1:6,]
```

```
## 
## TCGA-2A-A8VL-01A a8ad4b62-68e8-4d56-893e-e247a3099d94.mirbase21.isoforms.quanti
## TCGA-2A-A8VO-01A 22302d39-da19-4bfd-b4d8-aa951b9451a1.mirbase21.isoforms.quanti
## TCGA-2A-A8VT-01A de5cc4c2-2709-4bbe-8777-9d8e9cd56246.mirbase21.isoforms.quanti
## TCGA-2A-A8VV-01A f3402505-e2c1-4720-a9f2-f39105ad0327.mirbase21.isoforms.quanti
## TCGA-2A-A8VX-01A c2a2d423-e481-4821-9970-5e93d7d4442b.mirbase21.isoforms.quanti
## TCGA-2A-A8W1-01A b1a5f1a4-a95a-4770-a234-709c4e9da1fe.mirbase21.isoforms.quanti
##                                               file_id      patient
## TCGA-2A-A8VL-01A a0b6cbc1-43fa-4bed-83e8-917794158b98 TCGA-2A-A8VL
## TCGA-2A-A8VO-01A addea5e5-5b25-417c-bbb2-00438b8da4c6 TCGA-2A-A8VO
## TCGA-2A-A8VT-01A 7a337162-08ee-4600-96f5-79fed7b68898 TCGA-2A-A8VT
## TCGA-2A-A8VV-01A fc64fdd9-b679-4a97-bf5e-d757b64b252c TCGA-2A-A8VV
## TCGA-2A-A8VX-01A 8387f768-8d31-4ffa-88ae-dae0ef11b2fb TCGA-2A-A8VX
## TCGA-2A-A8W1-01A cb18f79c-41d4-4bb9-af6a-28e35b6a4470 TCGA-2A-A8W1
##                           sample     submitter_id
## TCGA-2A-A8VL-01A TCGA-2A-A8VL-01 TCGA-2A-A8VL-01A
## TCGA-2A-A8VO-01A TCGA-2A-A8VO-01 TCGA-2A-A8VO-01A
## TCGA-2A-A8VT-01A TCGA-2A-A8VT-01 TCGA-2A-A8VT-01A
## TCGA-2A-A8VV-01A TCGA-2A-A8VV-01 TCGA-2A-A8VV-01A
## TCGA-2A-A8VX-01A TCGA-2A-A8VX-01 TCGA-2A-A8VX-01A
## TCGA-2A-A8W1-01A TCGA-2A-A8W1-01 TCGA-2A-A8W1-01A
##                           entity_submitter_id  sample_type gender
## TCGA-2A-A8VL-01A TCGA-2A-A8VL-01A-21R-A37H-13 PrimaryTumor   male
## TCGA-2A-A8VO-01A TCGA-2A-A8VO-01A-11R-A37H-13 PrimaryTumor   male
## TCGA-2A-A8VT-01A TCGA-2A-A8VT-01A-11R-A37H-13 PrimaryTumor   male
## TCGA-2A-A8VV-01A TCGA-2A-A8VV-01A-11R-A37H-13 PrimaryTumor   male
## TCGA-2A-A8VX-01A TCGA-2A-A8VX-01A-11R-A37H-13 PrimaryTumor   male
## TCGA-2A-A8W1-01A TCGA-2A-A8W1-01A-11R-A37H-13 PrimaryTumor   male
##                  age_at_diagnosis tumor_stage tumor_grade days_to_death
## TCGA-2A-A8VL-01A            18658        <NA>        <NA>            NA
## TCGA-2A-A8VO-01A            20958        <NA>        <NA>            NA
## TCGA-2A-A8VT-01A            17365        <NA>        <NA>            NA
## TCGA-2A-A8VV-01A            19065        <NA>        <NA>            NA
## TCGA-2A-A8VX-01A            25904        <NA>        <NA>            NA
## TCGA-2A-A8W1-01A            19964        <NA>        <NA>            NA
##                  days_to_last_follow_up vital_status project_id
## TCGA-2A-A8VL-01A                      621        alive  TCGA-PRAD
## TCGA-2A-A8VO-01A                     1701        alive  TCGA-PRAD
## TCGA-2A-A8VT-01A                     1373        alive  TCGA-PRAD
## TCGA-2A-A8VV-01A                      671        alive  TCGA-PRAD
## TCGA-2A-A8VX-01A                     1378        alive  TCGA-PRAD
## TCGA-2A-A8W1-01A                      112        alive  TCGA-PRAD
```

4.2 Filter samples

4.2.1 Filter duplicated samples

Only one sample would be kept if the sample had been sequenced more than once by `gdcFilterDuplicate()`.

```
####### Filter duplicated samples in RNAseq metadata #######
metaMatrix.RNA <- gdcFilterDuplicate(metaMatrix.RNA)
```

```
## Removed 3 samples
```

```
####### Filter duplicated samples in miRNAs metadata #######
metaMatrix.MIR <- gdcFilterDuplicate(metaMatrix.MIR)
```

```
## Removed 4 samples
```

4.2.2 Filter non-Primary Tumor and non-Solid Tissue Normal samples

Samples that are neither Primary Tumor (code: 01) nor Solid Tissue Normal (code: 11) would be filtered out by `gdcFilterSampleType()`.

```
####### Filter non-Primary Tumor and non-Solid Tissue Normal samples in RNAseq meta
metaMatrix.RNA <- gdcFilterSampleType(metaMatrix.RNA)
```

```
## Removed 1 samples
```

```
####### Filter non-Primary Tumor and non-Solid Tissue Normal samples in miRNAs meta
metaMatrix.MIR <- gdcFilterSampleType(metaMatrix.MIR)
```

```
## Removed 1 samples
```

4.3 Merge data

- `gdcRNAMerge()` merges raw counts data of RNAseq to a single expression matrix with rows are *Ensembl id* and columns are *samples*. Total read counts for 5p and 3p strands of miRNAs can be processed from isoform quantification files and then merged to a single expression matrix with rows are *miRBase v21 identifiers* and columns are *samples*.
- `gdcClinicalMerge()` merges clinical data to a dataframe with rows are *patient id* and columns are *clinical traits*. If `key.info=TRUE`, only those most commonly used clinical traits will be reported, otherwise, all the clinical information will be reported.

4.3.1 Merge RNAseq/miRNAs data

```
####### Merge RNAseq data #######
rnaMatrix <- gdcRNAMerge(metadata  = metaMatrix.RNA,
                         path      = 'TCGA-PRAD/RNAseq/',
                         data.type = 'RNAseq')
```

```
## ############### Merging RNAseq data ################
## ### This step may take a few minutes ###
## Number of samples: 547
## Number of genes: 60483
```

```
rnaMatrix[1:6,1:6]
```

```
##                 TCGA-2A-A8VL-01 TCGA-2A-A8VO-01 TCGA-2A-A8VT-01
## ENSG00000000003            2867            1667            3140
## ENSG00000000005               6               0               0
## ENSG00000000419            1354             888            1767
## ENSG00000000457             956             580            2163
## ENSG00000000460             119              91             305
## ENSG00000000938             159             171             228
##                 TCGA-2A-A8VV-01 TCGA-2A-A8VX-01 TCGA-2A-A8W1-01
## ENSG00000000003            3996            4869            2172
## ENSG00000000005              44               1               0
## ENSG00000000419            1408            1171            1593
## ENSG00000000457            1494             908             794
## ENSG00000000460             175             121             166
## ENSG00000000938             172              64             161
```

```
####### Merge miRNAs data #######
mirMatrix <- gdcRNAMerge(metadata  = metaMatrix.MIR,
                         path      = 'TCGA-PRAD/miRNAs/',
                         data.type = 'miRNAs')
```

```
## ############### Merging miRNAs data ###############
## Number of samples: 546
## Number of miRNAs: 2588
```

```
mirMatrix[1:6,1:6]
```

```
##                 TCGA-2A-A8VL-01 TCGA-2A-A8VO-01 TCGA-2A-A8VT-01
## hsa-let-7a-5p            130022           77195          170937
## hsa-let-7a-3p               133              84              91
## hsa-let-7a-2-3p              18              10              13
## hsa-let-7b-5p             68276           19131           36009
## hsa-let-7b-3p                78              30              55
## hsa-let-7c-5p             43015           22490           14099
##                  TCGA-2A-A8VV-01 TCGA-2A-A8VX-01 TCGA-2A-A8W1-01
## hsa-let-7a-5p             247370           73705           50261
## hsa-let-7a-3p                104              59              39
## hsa-let-7a-2-3p               13               3               4
## hsa-let-7b-5p              58349           17404            6663
## hsa-let-7b-3p                 73              19              18
## hsa-let-7c-5p              36248            9694           11759
```

4.3.2 Merge clinical data

```
####### Merge clinical data #######
clinicalDa <- gdcClinicalMerge(path = 'TCGA-PRAD/Clinical/', key.info = TRU
```

```
## ############### Merging Clinical data ###############
```

```
clinicalDa[1:6,5:10]
```

```
##              clinical_stage clinical_T clinical_N clinical_M
## TCGA-EJ-5510             NA        T1c         NA         M0
## TCGA-HC-8260             NA         NA         NA         M0
## TCGA-Y6-A8TL             NA        T2a         NA         NA
## TCGA-V1-A8X3             NA        T1c         NA         M0
## TCGA-VP-A87J             NA        T2a         NA         M0
## TCGA-KK-A6DY             NA        T1c         NA         M0
##              gleason_grading gleason_score
## TCGA-EJ-5510            7433             7
## TCGA-HC-8260             734             7
## TCGA-Y6-A8TL             633             6
## TCGA-V1-A8X3             734             7
## TCGA-VP-A87J             734             7
## TCGA-KK-A6DY             734             7
```

4.4 TMM normalization and voom transformation

It has repeatedly shown that normalization is a critical way to ensure accurate estimation and detection of differential expression (DE) by removing systematic technical effects that occur in the data[@robinson_scaling_2010]. TMM normalization is a simple and effective method for estimating relative RNA production levels from RNA-seq data. Voom is moreover faster and more convenient than existing RNA-seq methods, and converts RNA-seq data into a form that can be analyzed using similar tools as for microarrays[@law_voom:_2014].

By running `gdcVoomNormalization()` function, raw counts data would be normalized by TMM method implemented in edgeRref and further transformed by the voom method provided in limmaref. Low expression genes (logcpm < 1 in more than half of the samples) will be filtered out by default. All the genes can be kept by setting `filter=TRUE` in the `gdcVoomNormalization()`.

```
####### RNAseq data #######
rnaExpr <- gdcVoomNormalization(counts = rnaMatrix, filter = FALSE)
rnaExpr[1:6,1:6]
```

```
##                 TCGA-2A-A8VL-01 TCGA-2A-A8VO-01 TCGA-2A-A8VT-01
## ENSG00000000003        5.891004        5.469541        5.675430
## ENSG00000000005       -2.894134       -6.233930       -6.941348
## ENSG00000000419        4.808971        4.561298        4.846146
## ENSG00000000457        4.307047        3.947222        5.137803
## ENSG00000000460        1.306293        1.281770        2.313680
## ENSG00000000938        1.722839        2.188135        1.894702
##                 TCGA-2A-A8VV-01 TCGA-2A-A8VX-01 TCGA-2A-A8W1-01
## ENSG00000000003       6.3329382       6.6613451        5.612615
## ENSG00000000005      -0.1558497      -5.0032503       -6.472525
## ENSG00000000419       4.8283607       4.6059284        5.165458
## ENSG00000000457       4.9138640       4.2391299        4.161378
## ENSG00000000460       1.8237441       1.3365997        1.906853
## ENSG00000000938       1.7988694       0.4230145        1.862865
```

```
####### miRNAs data #######
mirExpr <- gdcVoomNormalization(counts = mirMatrix, filter = FALSE)
mirExpr[1:6,1:6]
```

```
##                  TCGA-2A-A8VL-01 TCGA-2A-A8VO-01 TCGA-2A-A8VT-01
## hsa-let-7a-5p           14.676762       14.246607       15.773276
## hsa-let-7a-3p            4.749056        4.411257        4.905866
## hsa-let-7a-2-3p          1.897814        1.402695        2.145054
## hsa-let-7b-5p           13.747462       12.234040       13.526256
## hsa-let-7b-3p            3.982981        2.941115        4.184582
## hsa-let-7c-5p           13.080929       12.467406       12.173523
##                  TCGA-2A-A8VV-01 TCGA-2A-A8VX-01 TCGA-2A-A8W1-01
## hsa-let-7a-5p           15.705812      14.8712423       14.456561
## hsa-let-7a-3p            4.496858       4.5965754        4.143175
## hsa-let-7a-2-3p          1.544386       0.5091125        1.009320
## hsa-let-7b-5p           13.621931      12.7889304       11.541459
## hsa-let-7b-3p            3.989171       2.9871598        3.048848
## hsa-let-7c-5p           12.935132      11.9447084       12.360934
```

5 Differential gene expression analysis

`gdcDEAnalysis()`, a convenience wrapper, provides three widely used methods limmaref, edgeRref, and DESeq2ref to identify differentially expressed genes (DEGs) or miRNAs between any two groups defined by users. Note that DESeq2ref maybe slow with a single core. Multiple cores can be specified with the `nCore` argument if DESeq2ref is in use. Users are encouraged to consult the vignette of each method for more detailed information.

5.1 DE analysis

```
DEGAll <- gdcDEAnalysis(counts     = rnaMatrix,
                        group      = metaMatrix.RNA$sample_type,
                        comparison = 'PrimaryTumor-SolidTissueNormal',
                        method     = 'limma')
DEGAll[1:6,]
```

```
##                   symbol          group     logFC   AveExpr         t
## ENSG00000187699 C2orf88 protein_coding -2.657180 1.5056478 -19.46636
## ENSG00000176928   GCNT4 protein_coding -2.248112 0.5798701 -18.39206
## ENSG00000118298    CA14 protein_coding -2.630802 0.4748363 -17.57925
## ENSG00000103485    QPRT protein_coding -2.147259 1.9897483 -17.32704
## ENSG00000109667  SLC2A9 protein_coding -1.869863 1.6079446 -17.21612
## ENSG00000164764  SBSPON protein_coding -2.333725 2.5270242 -17.17468
##                       PValue          FDR        B
## ENSG00000187699 1.453473e-64 2.259715e-60 136.1299
## ENSG00000176928 3.303402e-59 2.567900e-55 123.6779
## ENSG00000118298 3.348976e-55 1.735551e-51 114.6210
## ENSG00000103485 5.729814e-54 2.227035e-50 111.9647
## ENSG00000109667 1.990189e-53 6.188294e-50 110.6756
## ENSG00000164764 3.166847e-53 8.205828e-50 110.2957
```

5.2 Report DE genes/miRNAs

All DEGs, DE long non-coding genes, DE protein coding genes and DE miRNAs could be reported separately by setting `geneType` argument in `gdcDEReport()`. Gene symbols and biotypes based on the Ensembl 90 annotation are reported in the output.

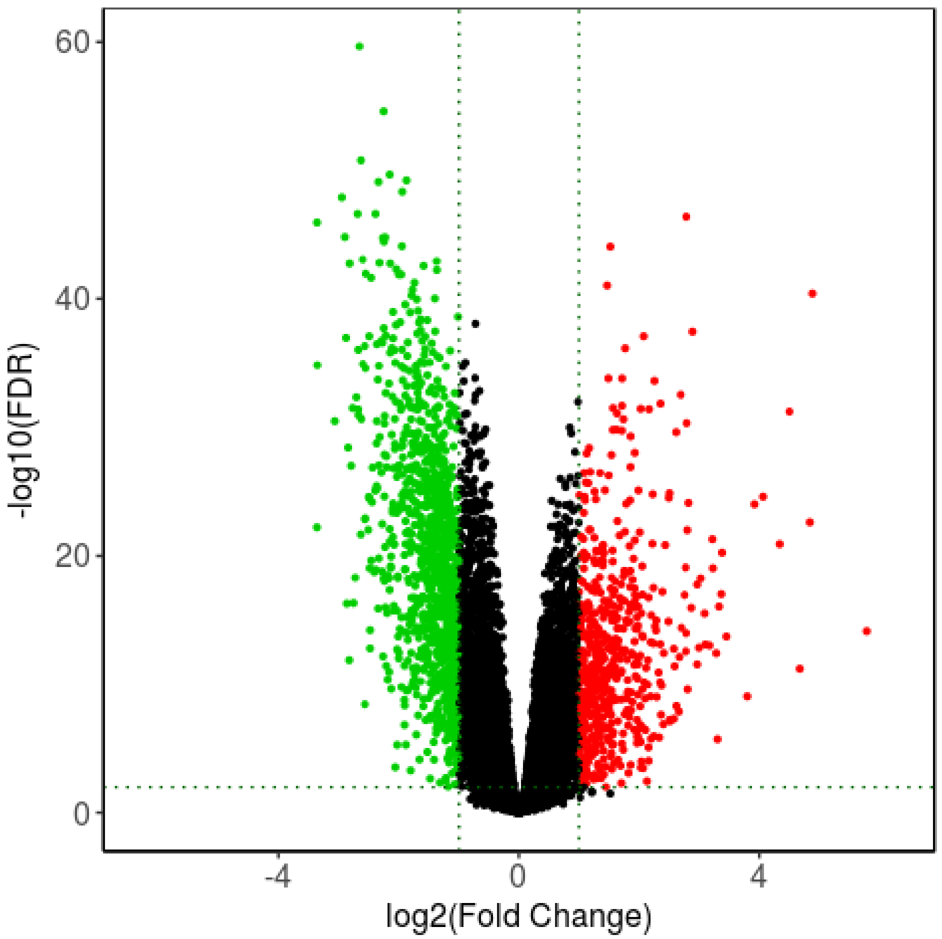
```
### All DEGs
deALL <- gdcDEReport(deg = DEGAll, gene.type = 'all')

#### DE long-noncoding
deLNC <- gdcDEReport(deg = DEGAll, gene.type = 'long_non_coding')

#### DE protein coding genes
dePC <- gdcDEReport(deg = DEGAll, gene.type = 'protein_coding')
```

5.3 DEG visualization

Volcano plot and bar plot are used to visualize DE analysis results in different manners by `gdcVolcanoPlot()` and `gdcBarPlot()` functions, respectively . Hierarchical clustering on the expression matrix of DEGs can be analyzed and plotted by the `gdcHeatmap()` function.

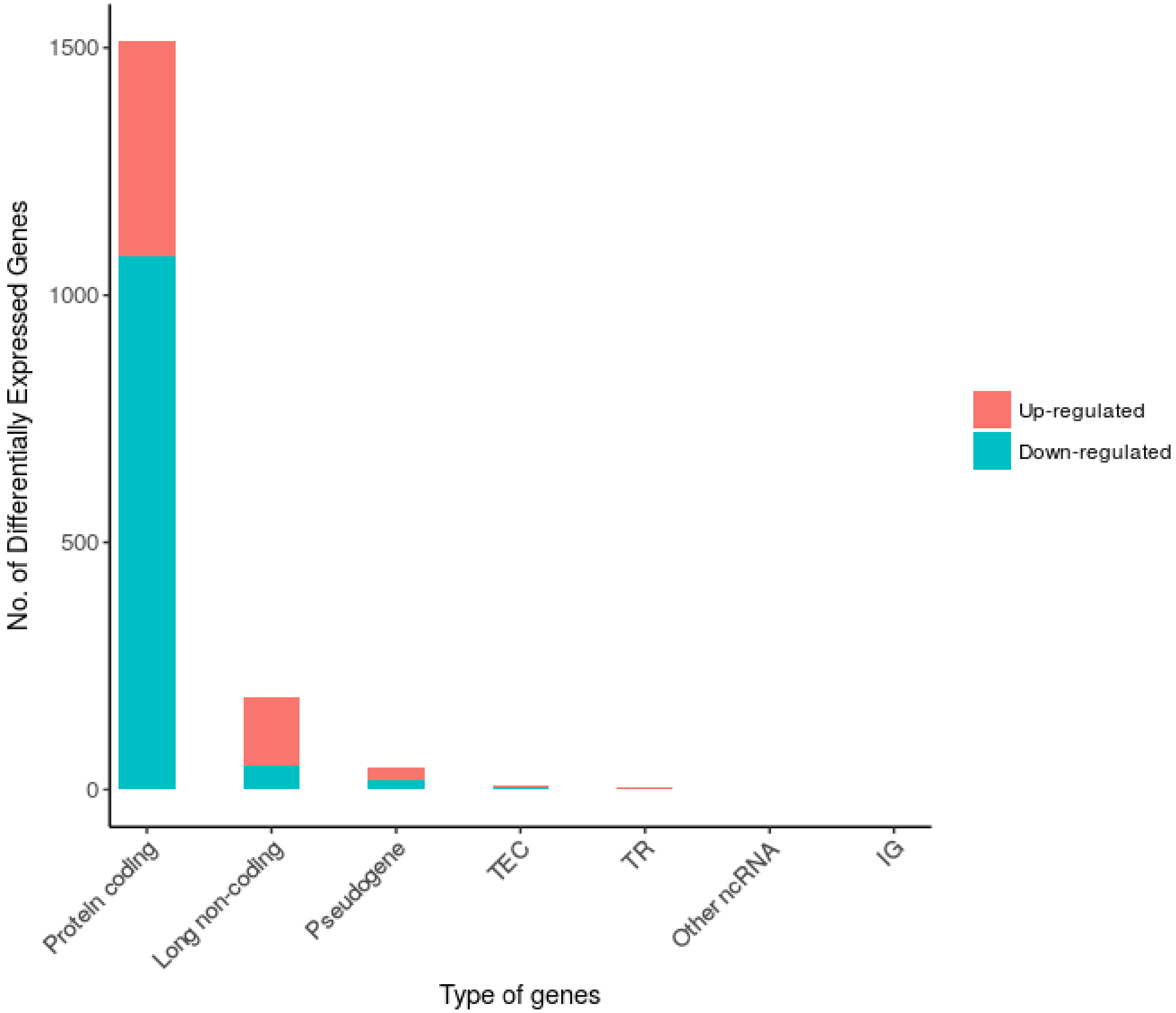
5.3.1 Volcano plot

```
gdcVolcanoPlot(DEGAll)
```



5.3.2 Barplot

```
gdcBarPlot(deg = deALL, angle = 45, data.type = 'RNAseq')
```



5.3.3 Heatmap

Heatmap is generated based on the `heatmap.2()` function in gplots package.

```
degName = rownames(deALL)
gdcHeatmap(deg.id = degName, metadata = metaMatrix.RNA, rna.expr = rnaExpr)
```

6 Competing endogenous RNAs network analysis

Three criteria are used to determine the competing endogenous interactions between lncRNA-mRNA pairs:

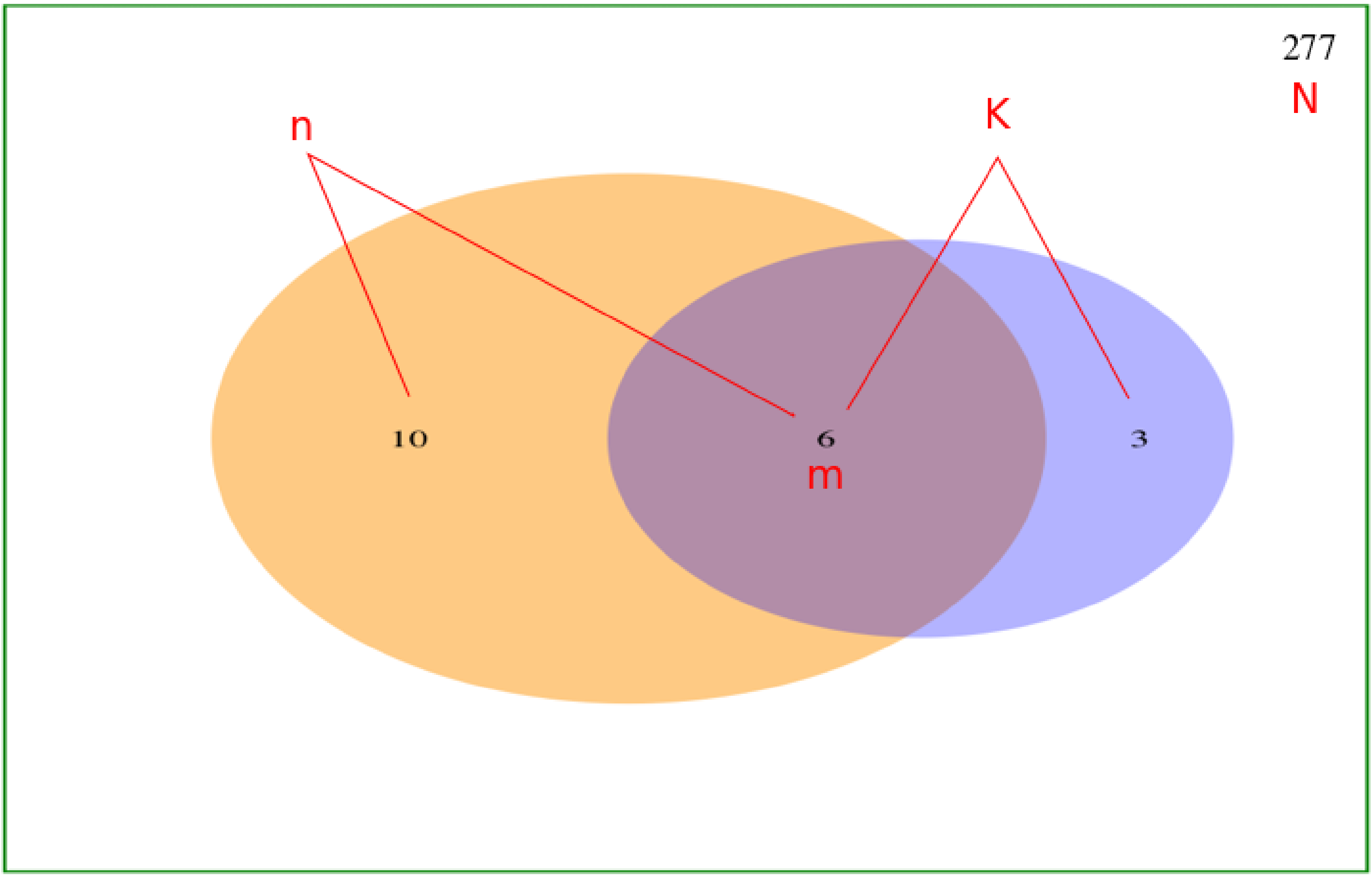
- The lncRNA and mRNA must share significant number of miRNAs
- Expression of lncRNA and mRNA must be positively correlated
- Those common miRNAs should play similar roles in regulating the expression of lncRNA and mRNA

6.1 Hypergeometric test

Hypergenometric test is performed to test whether a lncRNA and mRNA share many miRNAs significantly.

A newly developed algorithm **spongeScan**ref is used to predict MREs in lncRNAs acting as ceRNAs. Databases such as **starBase v2.0**ref, **miRcode**ref and **mirTarBase release 7.0**ref are used to collect predicted and experimentally validated miRNA-mRNA and/or miRNA-lncRNA interactions. Gene IDs in these databases are updated to the latest Ensembl 90 annotation of human genome and miRNAs names are updated to the new release miRBase 21 identifiers. Users can also provide their own datasets of miRNA-lncRNA and miRNA-mRNA interactions.

The figure and equation below illustrate how the hypergeometric test works



$$p = 1 - \sum_{k=0}^{m} \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

here m is the number of shared miRNAs, N is the total number of miRNAs in the database, n is the number of miRNAs targeting the lncRNA, K is the number of miRNAs targeting the protein coding gene.

6.2 Pearson correlation analysis

Pearson correlation coefficient is a measure of the strength of a linear association between two variables. As we all know, miRNAs are negative regulators of gene expression. If more common miRNAs are occupied by a lncRNA, less of them will bind to the target mRNA, thus increasing the expression level of mRNA. So expression of the lncRNA and mRNA in a ceRNA pair should be positively correlated.

6.3 Regulation pattern analysis

Two methods are used to measure the regulatory role of miRNAs on the lncRNA and mRNA:

- Regulation similarity

We defined a measurement *regulation similarity score* to check the similarity between miRNAs-lncRNA expression correlation and miRNAs-mRNA expression correlation.

$$Regulation\ similarity\ score = 1 - \frac{1}{M}\sum_{k=1}^{M}[\frac{|corr(m_k,l) - corr(m_k,g)|}{|corr(m_k,l)| + |corr(m_k,g)|}]^M$$

where M is the total number of shared miRNAs, k is the kth shared miRNAs, $corr(m_k,l)$ and $corr(m_k,g)$ represents the Pearson correlation between the kth miRNA and lncRNA, the kth miRNA and mRNA, respectively

- Sensitivity correlation

Sensitivity correlation is defined by Paci et al.ref to measure if the correlation between a lncRNA and mRNA is mediated by a miRNA in the lncRNA-miRNA-mRNA triplet. We take average of all triplets of a lncRNA-mRNA pair and their shared miRNAs as the sensitivity correlation between a selected lncRNA and mRNA.

$$Sensitivity\ correlation = corr(l,g) - \frac{1}{M}\sum_{k=1}^{M}\frac{corr(l,g) - corr(m_k,l)corr(m_k,g)}{\sqrt{1 - corr(m_k,l)^2}\sqrt{1 - corr(m_k,g)^2}}$$

where M is the total number of shared miRNAs, k is the kth shared miRNAs, $corr(l,g)$, $corr(m_k,l)$ and $corr(m_k,g)$ represents the Pearson correlation between the long non-coding RNA and the protein coding gene, the kth miRNA and lncRNA, the kth miRNA and mRNA, respectively

---

The hypergeometric test of shared miRNAs, expression correlation analysis of lncRNA-mRNA pair, and regulation pattern analysis of shared miRNAs are all implemented in the `gdcCEAnalysis()` function.

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

```
## Step 1/3: Hypergenometric test done !
## Step 2/3: Correlation analysis done !
## Step 3/3: Regulation pattern analysis done !
```

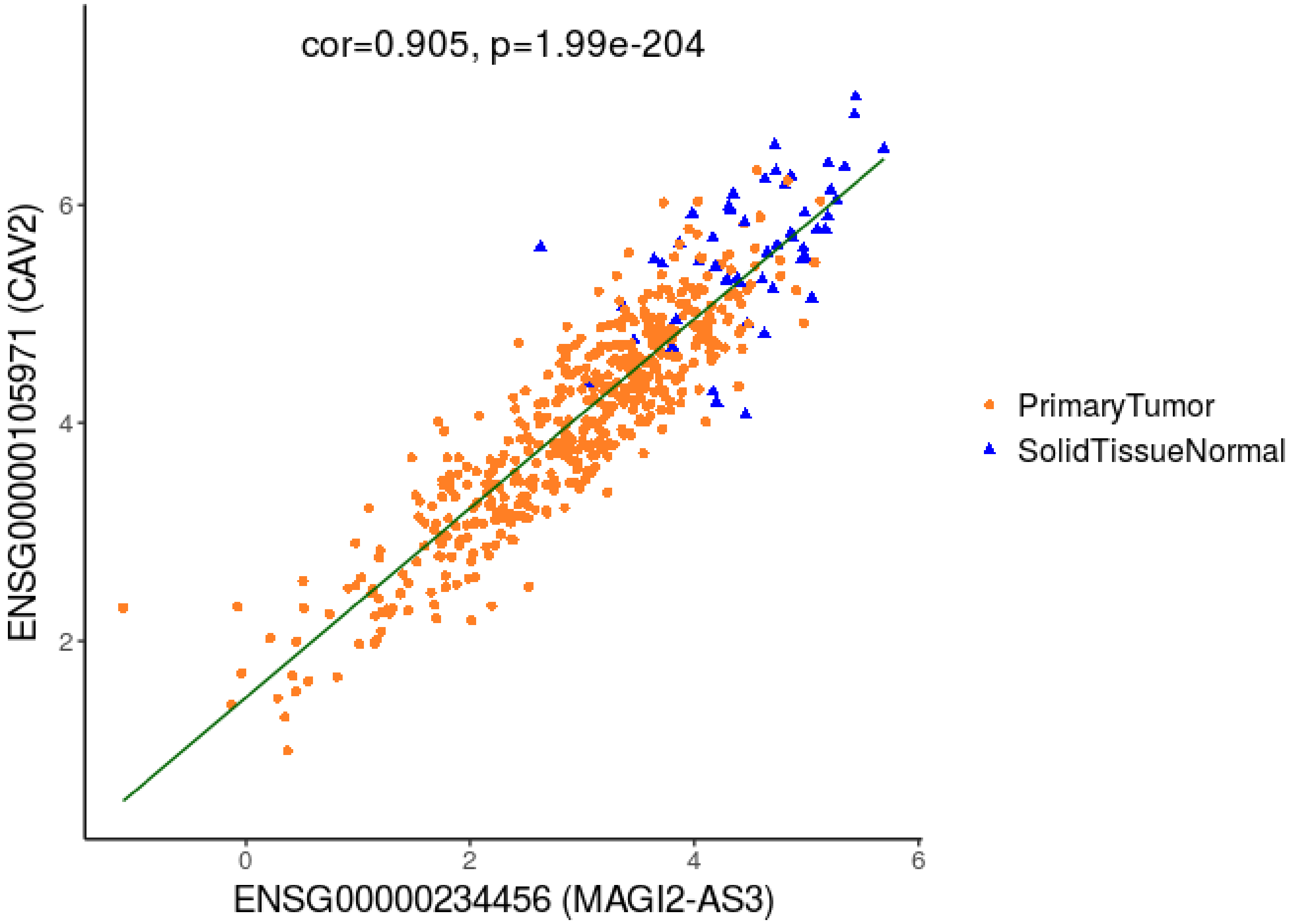
```
ceOutput <- ceOutput[order(ceOutput$regSim),]
ceOutput[1:6,]
```

```
##            lncRNAs           Genes Counts listTotal popHits popTotal
## 22 ENSG00000234456 ENSG00000163110      2         2      36      277
## 34 ENSG00000234456 ENSG00000043591      2         2      51      277
## 37 ENSG00000234456 ENSG00000119547      2         2      71      277
## 45 ENSG00000234456 ENSG00000112984      2         2       3      277
## 57 ENSG00000234456 ENSG00000184838      2         2      23      277
## 65 ENSG00000228223 ENSG00000129514      1         2      23      277
##      foldEnrichment          hyperPValue                         miRNAs
## 22 7.69444444444444    0.016480929210485 hsa-miR-374b-5p,hsa-miR-374a-5p
## 34 5.43137254901961   0.0333542614974101 hsa-miR-374b-5p,hsa-miR-374a-5p
## 37 3.90140845070423   0.0650081096635797 hsa-miR-374b-5p,hsa-miR-374a-5p
## 45 92.3333333333333 7.84806152880239e-05 hsa-miR-374b-5p,hsa-miR-374a-5p
## 57 12.0434782608696  0.00661853188929001 hsa-miR-374b-5p,hsa-miR-374a-5p
## 65 6.02173913043478    0.159446450060168                  hsa-miR-590-3p
##           cor corPValue regSim          sppc
## 22 -0.3384664 1.0000000      0  0.0006305722
## 34 -0.4330765 1.0000000      0  0.0012113510
## 37 -0.3662249 1.0000000      0 -0.0241567080
## 45 -0.4187818 1.0000000      0 -0.0297453228
## 57 -0.2210811 0.9999999      0 -0.0155415582
## 65 -0.3084449 1.0000000      0 -0.0061604421
```

6.4 ceRNAs visualization

6.4.1 Correlation plot

```
gdcCorPlot(gene1    = 'ENSG00000234456',
           gene2    = 'ENSG00000105971',
           rna.expr = rnaExpr,
           metadata = metaMatrix.RNA)
```

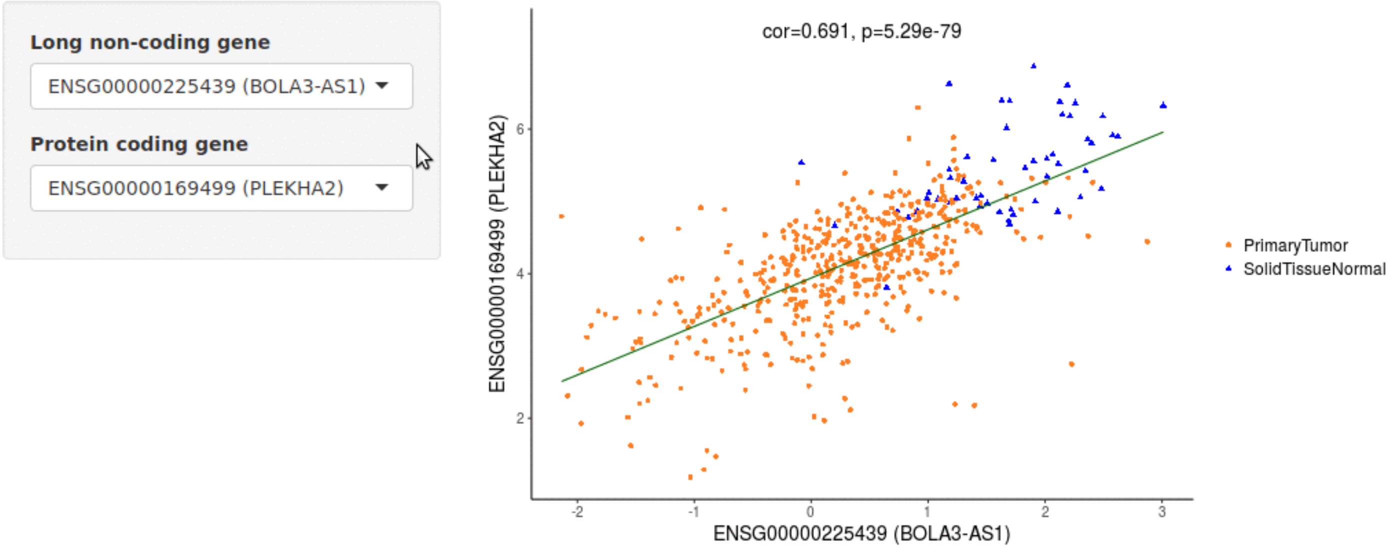


6.4.2 Correlation plot on a local webpage by shinyCorplot

Typing and running `gdcCorPlot()` for each pair of lncRNA-mRNA is bothering when multiple pairs are being interested in. `shinyCorPlot()` , a interactive plot function based on `shiny` package, can be easily operated by just clicking the genes in each drop down box (in the GUI window). By running `shinyCorPlot()` function, a local webpage would pop up and correlation plot between a lncRNA and mRNA would be automatically shown.

```
shinyCorPlot(gene1    = rownames(deLNC),
             gene2    = rownames(dePC),
            rna.expr = rnaExpr,
            metadata = metaMatrix.RNA)
```

Expression correlation of ce pairs



Long non-coding gene

ENSG00000225439 (BOLA3-AS1)

Protein coding gene

ENSG00000169499 (PLEKHA2)

6.4.3 Network visulization in Cytoscape

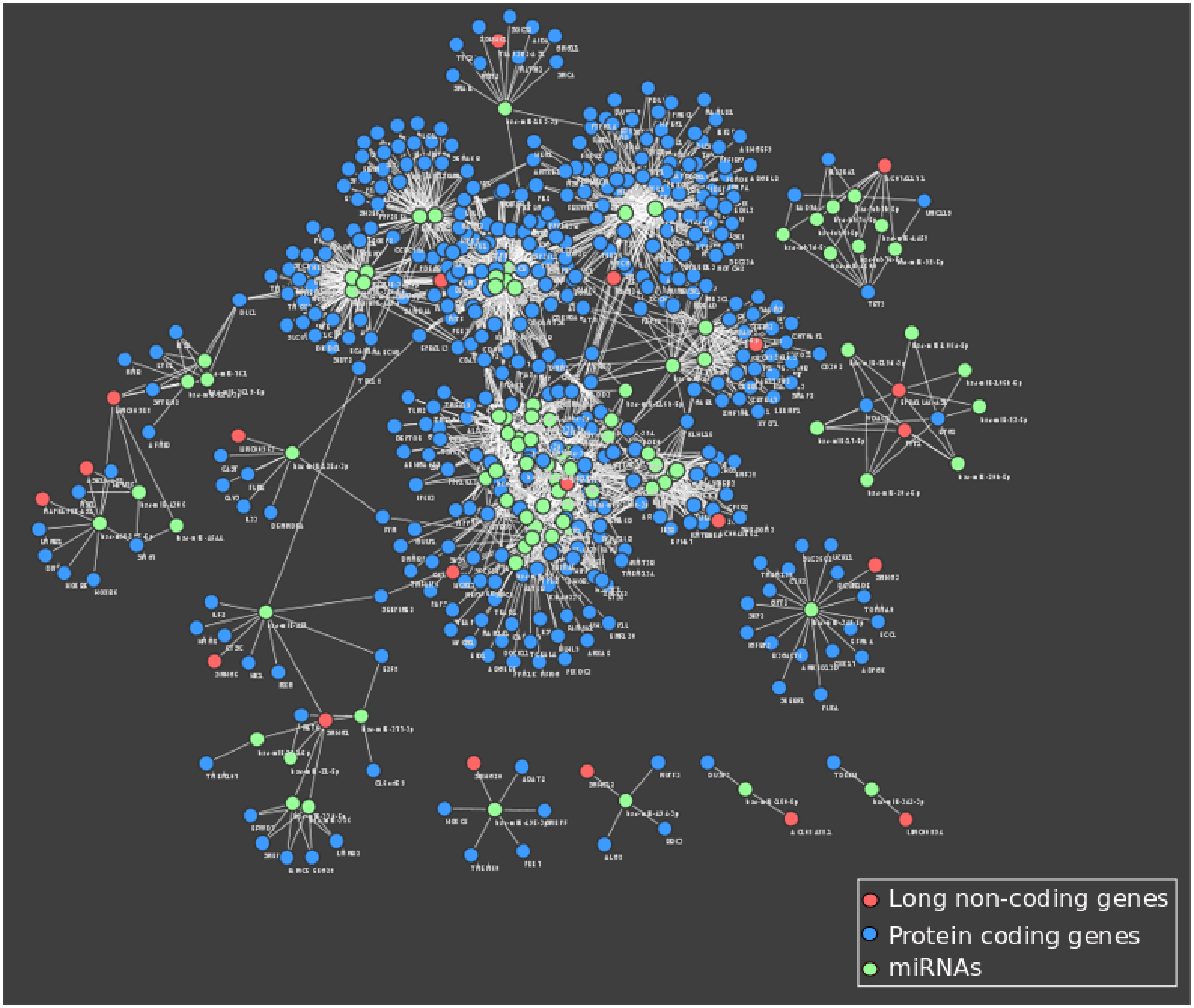
lncRNA-miRNA-mRNA interactions can be reported by the `gdcExportNetwork()` and visualized in **Cytoscape**.

```
ceOutput2 <- ceOutput[ceOutput$hyperPValue<0.01 & ceOutput$corPValue<0.01 & ceOut

edges <- gdcExportNetwork(ceNetwork = ceOutput2, net = 'edges')
edges[1:6,]
```

```
##          fromNode         toNode altNode1Name
## 1 ENSG00000234456 hsa-miR-374b-5p    MAGI2-AS3
## 2 ENSG00000234456 hsa-miR-374a-5p    MAGI2-AS3
## 3 ENSG00000245532   hsa-let-7i-5p        NEAT1
## 4 ENSG00000245532   hsa-let-7e-5p        NEAT1
## 5 ENSG00000245532   hsa-let-7g-5p        NEAT1
## 6 ENSG00000245532   hsa-let-7f-5p        NEAT1
```

```
nodes <- gdcExportNetwork(ceNetwork = ceOutput2, net = 'nodes')
nodes[1:6,]
```

```
##                 gene symbol type numInteractions
## 1 ENSG00000008300 CELSR3   pc               8
## 2 ENSG00000047597     XK   pc               2
## 3 ENSG00000065320   NTN1   pc               2
## 4 ENSG00000065534   MYLK   pc               2
## 5 ENSG00000066468  FGFR2   pc               2
## 6 ENSG00000075651   PLD1   pc               1
```



7 Univariate survival analysis

Two methods are provided to perform univariate survival analysis: Cox Proportional-Hazards (CoxPH) model and Kaplan Meier (KM) analysis based on the survival package. CoxPH model considers expression value as continous variable while KM analysis divides patients into high-expreesion and low-expression groups by a user-defined threshold such as median or mean. `gdcSurvivalAnalysis()` take a list of genes as input and report the hazard ratio, 95% confidence intervals, and test significance of each gene on overall survival.

7.1 CoxPH analysis

```
####### CoxPH analysis #######
survOutput <- gdcSurvivalAnalysis(gene     = rownames(deALL),
                                  method   = 'coxph',
                                  rna.expr = rnaExpr,
                                  metadata = metaMatrix.RNA)

head(survOutput[order(survOutput$pValue),])
```

```
##                      symbol       coef        HR   lower95    upper95
## ENSG00000156804     FBXO32 -0.9061689 0.4040693 0.2444365  0.6679526
## ENSG00000273478 AC099676.1  1.8426288 6.3131126 1.9864365 20.0637629
## ENSG00000069535       MAOB -0.4870443 0.6144398 0.4517982  0.8356304
## ENSG00000128298   BAIAP2L2  0.4950804 1.6406302 1.1837845  2.2737816
## ENSG00000255545 AP004608.1  0.7108727 2.0357671 1.2702956  3.2625066
## ENSG00000180447       GAS1 -0.6253213 0.5350895 0.3530306  0.8110367
##                       pValue
## ENSG00000156804 0.0004100575
## ENSG00000273478 0.0017880566
## ENSG00000069535 0.0019053414
## ENSG00000128298 0.0029472842
## ENSG00000255545 0.0031344643
## ENSG00000180447 0.0032084588
```

7.2 KM analysis

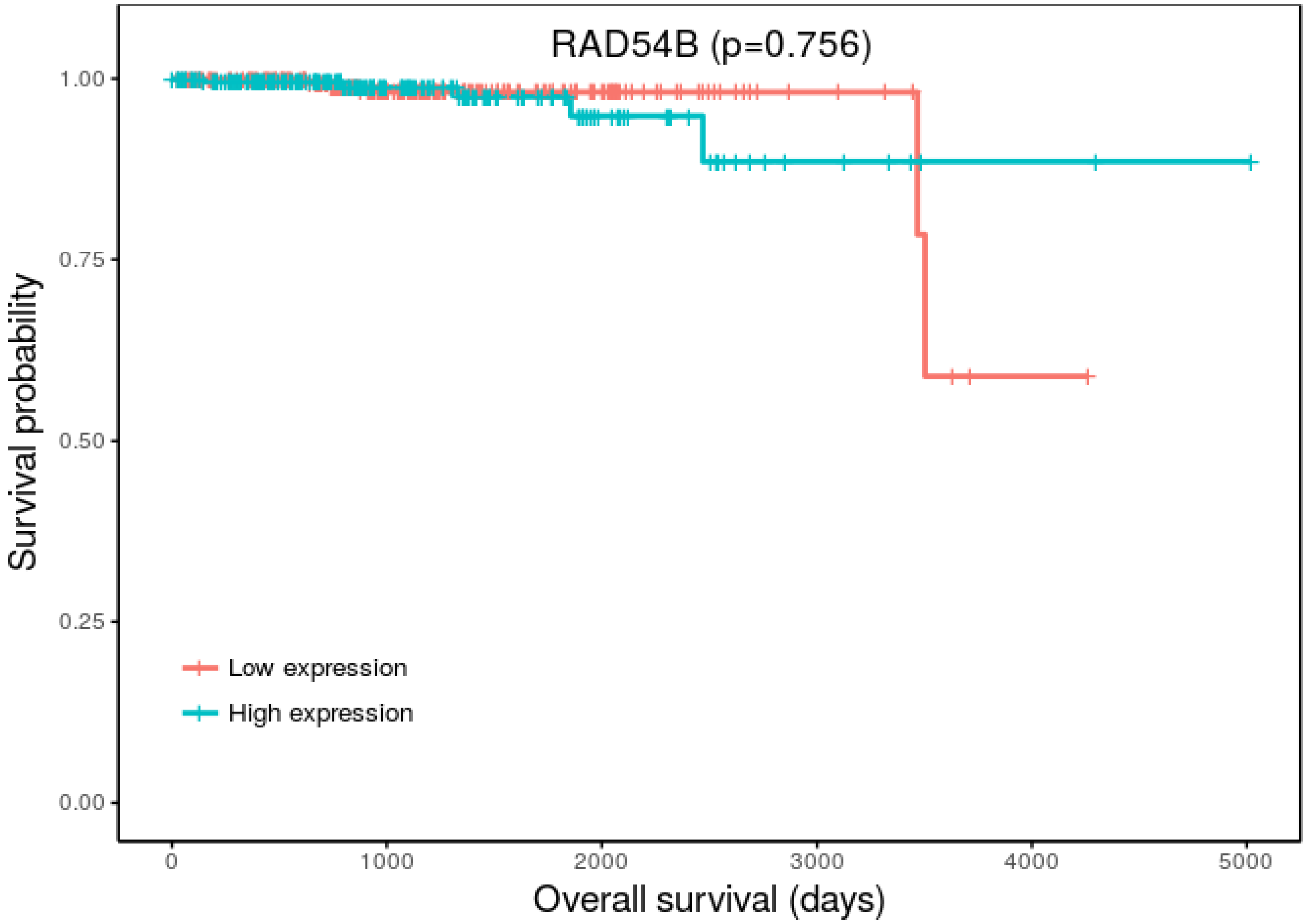
```
####### KM analysis #######
survOutput <- gdcSurvivalAnalysis(gene     = rownames(deALL),
                                  method   = 'KM',
                                  rna.expr = rnaExpr,
                                  metadata = metaMatrix.RNA,
                                  sep      = 'median')
```

7.3 KM analysis visualization

7.3.1 KM plot

KM survival curves are ploted using the `gdcKMPlot()` function which is based on the R package survminer.

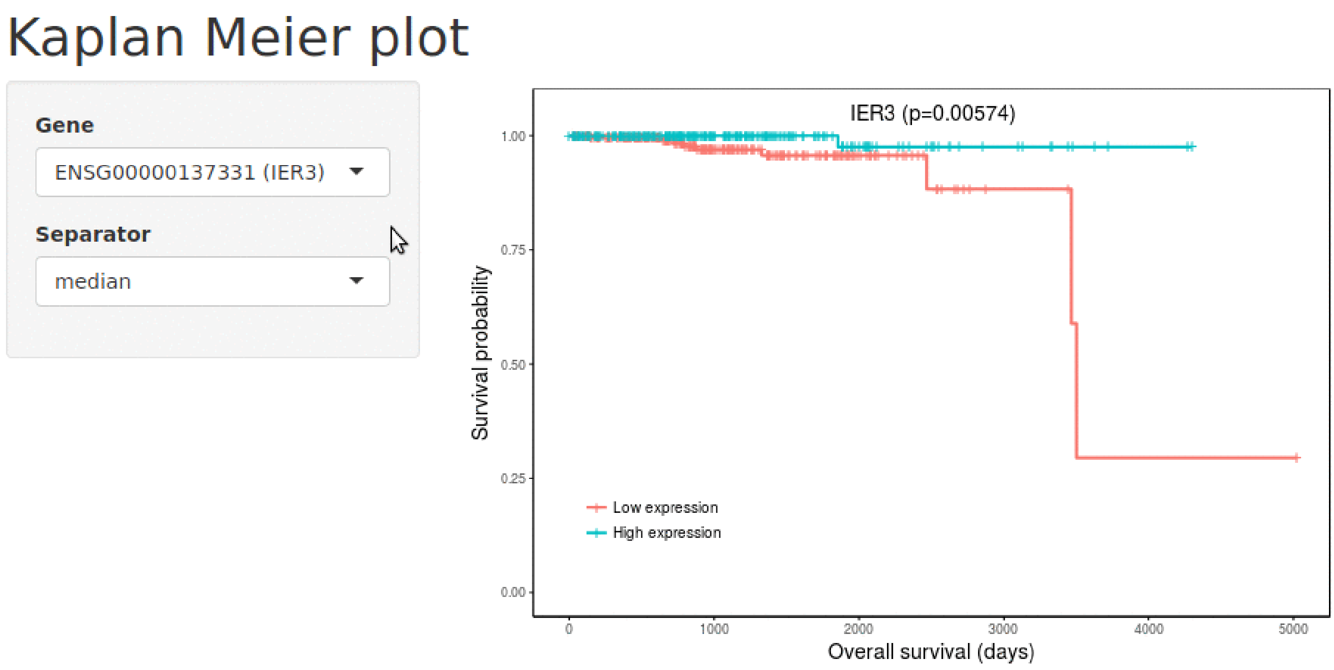
```
gdcKMPlot(gene     = 'ENSG00000197275',
          rna.expr = rnaExpr,
          metadata = metaMatrix.RNA,
          sep      = 'median')
```



7.3.2 KM plot on a local webpage by shinyKMPlot

The `shinyKMPlot()` function is also a simply `shiny` app which allow users view KM plots of all genes of interests on a local webpackage conveniently.

```
shinyKMPlot(gene = rownames(deALL), rna.expr = rnaExpr, metadata = metaMatri
```



Kaplan Meier plot

Gene

ENSG00000137331 (IER3)

Separator

median

8 Functional enrichment analysis

One of the main uses of the GO is to perform enrichment analysis on gene sets. For example, given a set of genes that are up-regulated under certain conditions, an enrichment analysis will find which GO terms are over-represented (or under-represented) using annotations for that gene set and pathway enrichment can also be applied afterwards.

8.1 GO, KEGG and DO analyses

`gdcEnrichAnalysis()` can perform Gene ontology (GO), Kyoto Encyclopedia of Genes and Genomes (KEGG) and Disease Ontology (DO) functional enrichment analyses of a list of genes simultaneously. GO and KEGG analyses are based on the R/Bioconductor packages clusterProfilierref and DOSEref. Redundant GO terms can be removed by specifying `simplify=TRUE` in the `gdcEnrichAnalysis()` function which uses the `simplify()` function in the clusterProfilierref package.

```
enrichOutput <- gdcEnrichAnalysis(gene = rownames(deALL), simplify = TRUE)
```

```
## ### This step may take a few minutes ###
## Step 1/5: BP analysis done!
## Step 2/5: CC analysis done!
## Step 3/5: MF analysis done!
## Step 4/5: KEGG analysis done!
## Step 5/5: DO analysis done!
```

```
terms <- c()
for (category in c('GO_BP', 'GO_CC', 'GO_MF', 'KEGG', 'DO')) {
  terms <- c(terms, which(enrichOutput$Category==category)[1:3])
}

enrichOutput[terms,]
```

```
##                                                                  Terms Counts
## 1                                        GO:0006936~muscle contraction     77
## 2                         GO:2000027~regulation of organ morphogenesis     47
## 3                      GO:0051146~striated muscle cell differentiation     56
## 63                                     GO:0031012~extracellular matrix     91
## 64                                        GO:0043292~contractile fiber     55
## 65                                               GO:0042383~sarcolemma     38
## 79                                GO:0005539~glycosaminoglycan binding     43
## 80                                         GO:0015267~channel activity     71
## 81                GO:0022803~passive transmembrane transporter activity     71
## 91                               hsa05414~Dilated cardiomyopathy (DCM)     25
## 92                          hsa05410~Hypertrophic cardiomyopathy (HCM)     22
## 93  hsa05412~Arrhythmogenic right ventricular cardiomyopathy (ARVC)     20
## 101                                         DOID:10283~prostate cancer     76
## 102                           DOID:3856~male reproductive organ cancer     76
## 103                                                  DOID:423~myopathy     69
##     GeneRatio   BgRatio       pValue          FDR foldEnrichment
## 1     77/1353 326/16447 9.881094e-18 4.882249e-14       2.871191
## 2     47/1353 185/16447 1.527429e-12 2.515676e-09       3.088268
## 3     56/1353 249/16447 2.588835e-12 2.741716e-09       2.733868
## 63    91/1431 425/17563 5.476425e-18 2.650589e-15       2.627916
## 64    55/1431 217/17563 1.472759e-14 2.376051e-12       3.110728
## 65    38/1431 124/17563 3.068355e-13 3.320284e-11       3.761153
## 79    43/1351 200/16514 3.287862e-09 2.689471e-06       2.628061
## 80    71/1351 450/16514 5.409995e-08 1.475125e-05       1.928603
## 81    71/1351 450/16514 5.409995e-08 1.475125e-05       1.928603
## 91     25/607   89/7174 4.416347e-08 1.245410e-05       3.319882
## 92     22/607   83/7174 8.615268e-07 1.139528e-04       3.132689
## 93     20/607   72/7174 1.212263e-06 1.139528e-04       3.282995
## 101    76/842  412/7577 3.902626e-06 2.993314e-03       1.659975
## 102    76/842  422/7577 9.701641e-06 3.720579e-03       1.620639
## 103    69/842  385/7577 2.987141e-05 5.727843e-03       1.612774
##
## 1
## 2
## 3
## 63  ENSG00000164764/ENSG00000095713/ENSG00000197565/ENSG00000154736/ENSG00000183
## 64
## 65
## 79
## 80
## 81
## 91
## 92
## 93
## 101
## 102
## 103
##
## 1
## 2
## 3
## 63  SBSPON/CRTAC1/COL4A6/ADAMTS5/CCBE1/NAV2/MAMDC2/TGFBR3/DPT/GPLD1/FGFR2/NDP/FG
## 64
## 65
## 79
## 80
## 81
## 91
## 92
## 93
## 101
## 102
## 103
##      Category
## 1       GO_BP
## 2       GO_BP
## 3       GO_BP
## 63      GO_CC
## 64      GO_CC
## 65      GO_CC
## 79      GO_MF
## 80     GO_MF
## 81     GO_MF
## 91      KEGG
## 92      KEGG
## 93      KEGG
## 101       DO
## 102       DO
## 103       DO
```

8.2 Enrichment visualization

The output generated by `gdcEnrichAnalysis()` can be used for visualization in the `gdcEnrichPlot()` function by specifying `type`, `category` and `numTerms` arguments.

8.2.1 GO barplot

```
gdcEnrichPlot(enrichOutput, type = 'bar', category = 'GO', num.terms = 10)
```

8.2.2 GO bubble plot

```
gdcEnrichPlot(enrichOutput, type='bubble', category='GO', num.terms = 10)
```

8.2.3 KEGG/DO barplot

```
gdcEnrichPlot(enrichment = enrichOutput,
              type       = 'bar',
              category   = 'KEGG',
              bar.color  = 'chocolate1',
              num.terms  = 20)
```

```
gdcEnrichPlot(enrichment = enrichOutput,
              type       = 'bar',
              category   = 'DO',
              bar.color  = 'dodgerblue',
              num.terms  = 20)
```

8.2.4 KEGG/DO bubble plot

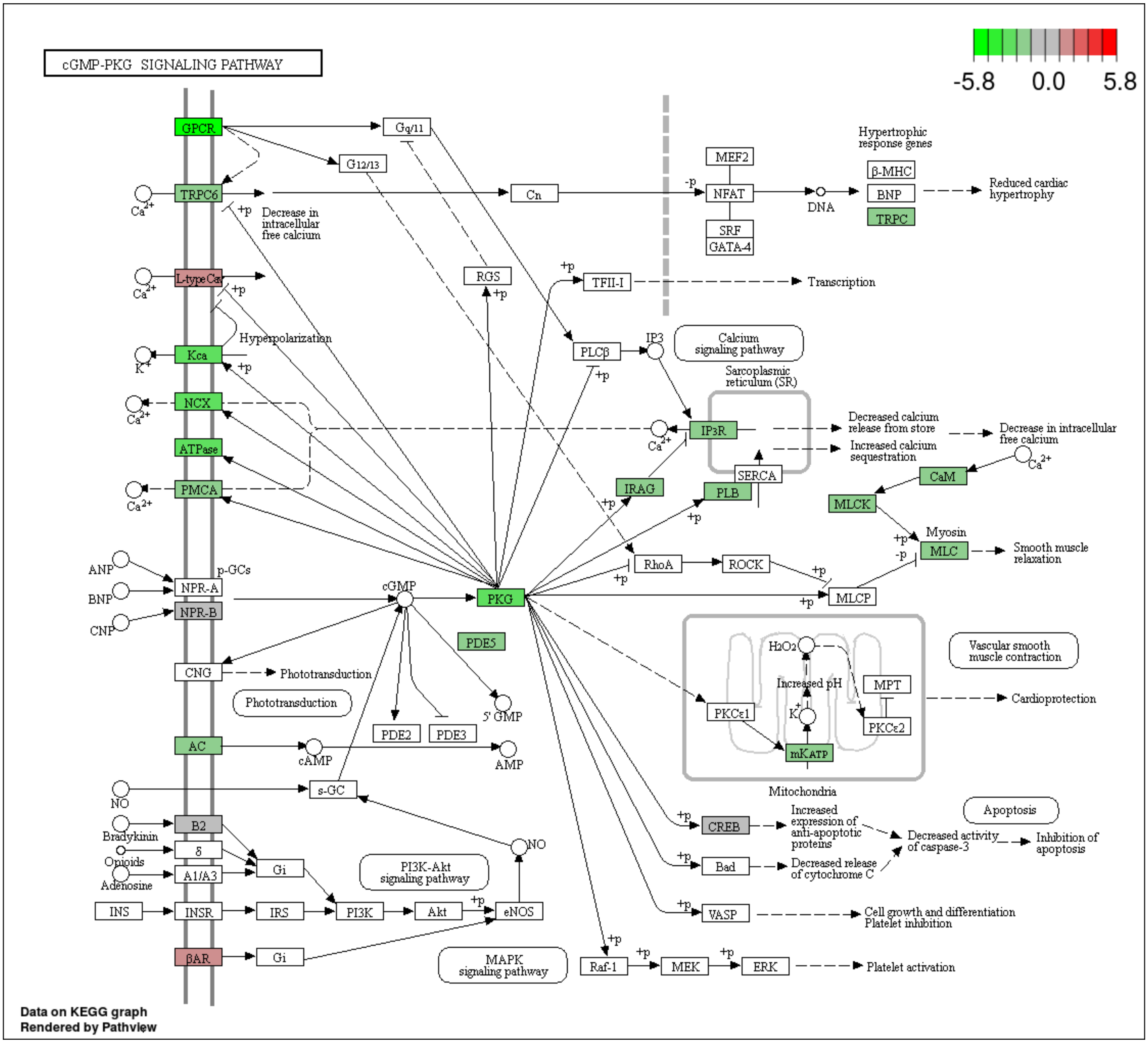
```
gdcEnrichPlot(=enrichOutput, category='KEGG',type = 'bubble', num.terms = 2
```

```
gdcEnrichPlot(enrichOutput, category='DO',type = 'bubble', num.terms = 20)
```

8.2.5 Pathview

Users can visualize a pathway map with `pathview()` function in the pathviewref package. It displays related many-genes-to-many-terms on 2-D view, shows by genes on BioCarta & KEGG pathway maps. Gradient colors can be used to indicate if genes are up-regulated or down-regulated.

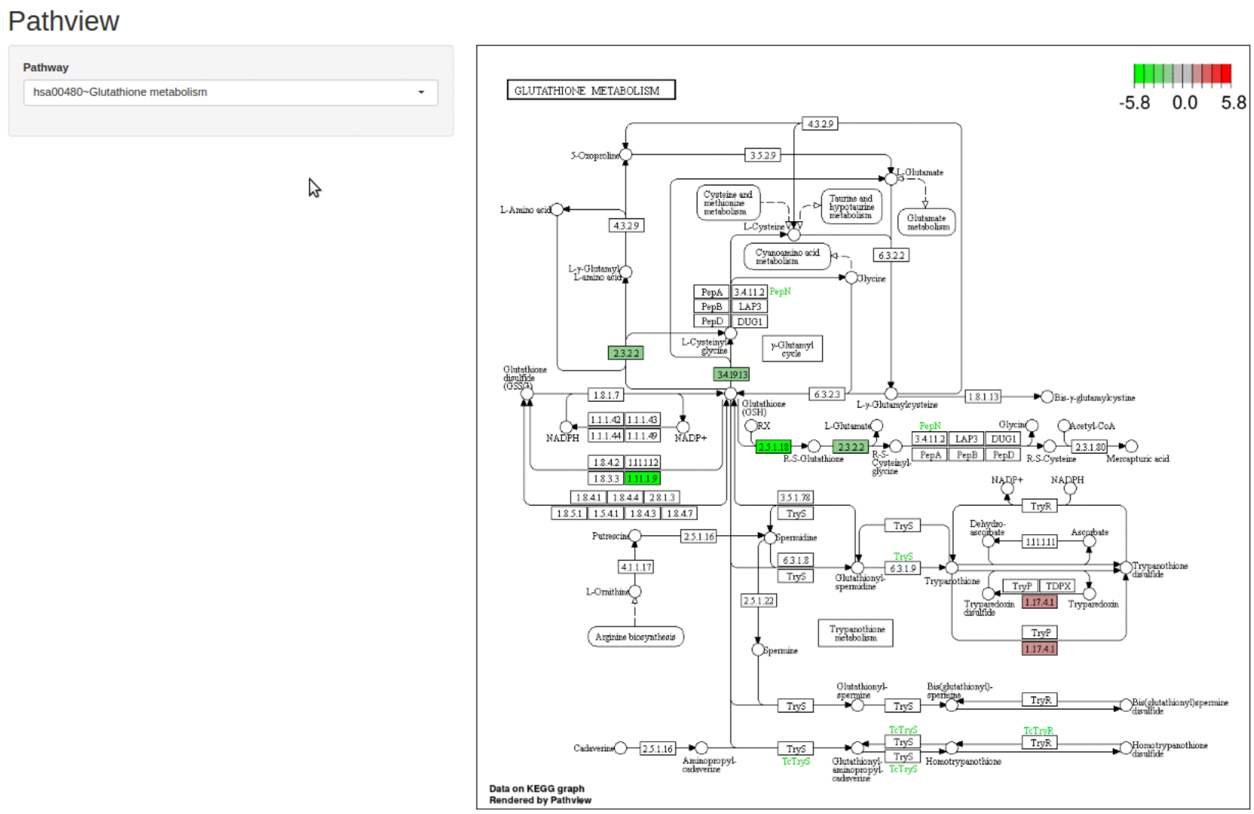
```
library(pathview)
deg <- deALL$logFC
names(deg) <- rownames(deALL)

hsa04022 <- pathview(gene.data   = deg,
                     pathway.id  = "hsa04022",
                     species     = "hsa",
                     gene.idtype = 'ENSEMBL',
                     limit       = list(gene=max(abs(geneList)), cpd=1))
```



8.2.6 View pathway maps on a local webpage by shinyPathview

`shinyPathview()` allows users view and download pathways of interests by simply selecting the pathway terms on a local webpage.

```
pathways <- as.character(enrichOutput$Terms[enrichOutput$Category=='KEGG'])

shinyPathview(deg, pathways = pathways, directory = 'pathview')
```



Pathview

Pathway: hsa00480~Glutathione metabolism

9 sessionInfo

```
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.1 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
```

```
## [1] GDCRNATools_0.99.0
## 
## loaded via a namespace (and not attached):
##   [1] colorspace_1.3-2           rjson_0.2.15              
##   [3] rprojroot_1.2              qvalue_2.4.2              
##   [5] htmlTable_1.9              XVector_0.12.1            
##   [7] GenomicRanges_1.24.3       base64enc_0.1-3           
##   [9] ggpubr_0.1.6               topGO_2.24.0              
##  [11] bit64_0.9-7                AnnotationDbi_1.34.4      
##  [13] splines_3.3.1              mnormt_1.5-5              
##  [15] GOSemSim_1.30.3            geneplotter_1.50.0        
##  [17] knitr_1.17                 Formula_1.2-2             
##  [19] jsonlite_1.5               broom_0.4.2               
##  [21] km.ci_0.5-2                annotate_1.50.1           
##  [23] cluster_2.0.6              GO.db_3.3.0               
##  [25] graph_1.50.0               shiny_1.0.5               
##  [27] backports_1.1.1            assertthat_0.2.0          
##  [29] Matrix_1.2-11              lazyeval_0.2.1            
##  [31] limma_3.28.21              acepack_1.4.1             
##  [33] htmltools_0.3.6            tools_3.3.1               
##  [35] bindrcpp_0.2               igraph_1.1.2              
##  [37] gtable_0.2.0               glue_1.2.0                
##  [39] reshape2_1.4.2             DO.db_2.9                 
##  [41] dplyr_0.7.4                Rcpp_0.12.13              
##  [43] Biobase_2.32.0             gdata_2.18.0              
##  [45] nlme_3.1-131               psych_1.7.8               
##  [47] stringr_1.2.0              mime_0.5                  
##  [49] clusterProfiler_3.0.5      gtools_3.5.0              
##  [51] XML_3.98-1.9               DOSE_2.10.7               
##  [53] org.Hs.eg.db_3.3.0         edgeR_3.14.0              
##  [55] zlibbioc_1.18.0            zoo_1.8-0                 
##  [57] scales_0.5.0               parallel_3.3.1            
##  [59] SummarizedExperiment_1.2.3 SparseM_1.77              
##  [61] RColorBrewer_1.1-2         yaml_2.1.14               
##  [63] memoise_1.1.0              gridExtra_2.3             
##  [65] KMsurv_0.1-5               ggplot2_2.2.1             
##  [67] biomaRt_2.28.0             rpart_4.1-11              
##  [69] latticeExtra_0.6-28        stringi_1.1.5             
##  [71] RSQLite_2.0                genefilter_1.54.2         
##  [73] S4Vectors_0.10.3           checkmate_1.8.5           
##  [75] caTools_1.17.1             BiocGenerics_0.18.0       
##  [77] BiocParallel_1.6.6         GenomeInfoDb_1.8.7        
##  [79] rlang_0.1.4                pkgconfig_2.0.1           
##  [81] matrixStats_0.52.2         bitops_1.0-6              
##  [83] evaluate_0.10.1            lattice_0.20-35           
##  [85] purrr_0.2.4                bindr_0.1
##  [87] labeling_0.3               htmlwidgets_0.9
##  [89] cmprsk_2.2-7               tidyselect_0.2.3
##  [91] bit_1.1-12                 GSEABase_1.34.1
##  [93] plyr_1.8.4                 magrittr_1.5
##  [95] DESeq2_1.12.4              R6_2.2.2
##  [97] IRanges_2.6.1              gplots_3.0.1
##  [99] Hmisc_4.0-3                DBI_0.7
## [101] foreign_0.8-69             prettydoc_0.2.0
## [103] survival_2.41-3            RCurl_1.95-4.8
## [105] nnet_7.3-12                tibble_1.3.4
## [107] survMisc_0.5.4             KernSmooth_2.23-15
## [109] rmarkdown_1.7              locfit_1.5-9.1
## [111] grid_3.3.1                 data.table_1.10.4-3
## [113] blob_1.1.0                 digest_0.6.12
## [115] xtable_1.8-2               tidyr_0.7.2
## [117] httpuv_1.3.5               stats4_3.3.1
## [119] munsell_0.4.3              survminer_0.4.0
```

10 Referneces