Funm6AViewer\_Usage

Songyao

2019/10/19

## 1. Installation

Funm6AViewer depends on GenomicFeatures, Guitar, trackViewer, DESeq2, STRINGdb, TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.eg.db R packages and please make sure they are installed before installing Funm6AViewer. An R version >= 3.6 is required.

Install the required packages

if (!requireNamespace("BiocManager", quietly = TRUE))   
 install.packages("BiocManager")  
  
BiocManager::install("GenomicFeatures")  
BiocManager::install("Guitar")  
BiocManager::install("trackViewer")  
BiocManager::install("DESeq2")  
BiocManager::install("STRINGdb")  
BiocManager::install("TxDb.Hsapiens.UCSC.hg19.knownGene")  
BiocManager::install("org.Hs.eg.db")

Install Funm6AViewer

if (!requireNamespace("devtools", quietly = TRUE))  
 install.packages("devtools")  
devtools::install\_github("NWPU-903PR/Funm6AViewer")

## 2. Data required

Funm6AViewer adopted FunDMDeep-m6A to idenify functional DmM genes which required 4 PPI networks. Then to run Funm6AViewer, users need to firstly download this data from <https://pan.baidu.com/s/1qOGG57OgxmrTwSbbBEeQ2w&shfl=sharepset>

## 3. One step usage of Funm6AViewer

funm6aviewer takes single base DmM sites information, gene DE information as input and output:

1. Functional DmM genes (FDmMGenes);
2. DmM sites distribution on RNA;
3. Counts of DmM sites on different RNA regions;
4. DmM sites and reads coverage on interested gene;
5. Function enrichment of FDmMGenes;
6. Context specific function of interested genes;
7. DmMGene’s MSB score along with DE score;
8. Network of FDmMGene’s MSB neighbors.

Following is an example to achieve these using funm6aviewer:

library(Funm6AViewer)

## Loading required package: GenomicFeatures

## Loading required package: BiocGenerics

## Loading required package: parallel

##   
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':  
##   
## clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
## clusterExport, clusterMap, parApply, parCapply, parLapply,  
## parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':  
##   
## IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, append, as.data.frame, basename, cbind,  
## colnames, dirname, do.call, duplicated, eval, evalq, Filter,  
## Find, get, grep, grepl, intersect, is.unsorted, lapply, Map,  
## mapply, match, mget, order, paste, pmax, pmax.int, pmin,  
## pmin.int, Position, rank, rbind, Reduce, rownames, sapply,  
## setdiff, sort, table, tapply, union, unique, unsplit, which,  
## which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##   
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:base':  
##   
## expand.grid

## Loading required package: IRanges

##   
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':  
##   
## windows

## Loading required package: GenomeInfoDb

## Loading required package: GenomicRanges

## Loading required package: AnnotationDbi

## Loading required package: Biobase

## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: TxDb.Hsapiens.UCSC.hg19.knownGene

## Loading required package: org.Hs.eg.db

##

Get input data:

dminfo <- system.file("extdata", "DMinfo\_toy.xls", package="Funm6AViewer")  
deinfo <- system.file("extdata", "DEinfo\_toy.xls", package="Funm6AViewer")  
  
dminfo <- read.table(dminfo, header = TRUE, stringsAsFactors = FALSE)  
deinfo <- read.delim(deinfo, header = TRUE, stringsAsFactors = FALSE)

dminfo contains the position annotation and log2 foldchange of DmM sites. It can be extracted from the result of DMDeep-m6A package using summarydmdeepm6A. Alternatively, users can use any other method to make it as the following formate:

head(dminfo)

## chr chromStart chromEnd name score strand log2fd  
## 1 chr1 155160832 155160833 4582 0.9137714 - 2.7950754  
## 2 chr1 171505224 171505225 23215 0.9431386 + 0.7589479  
## 3 chr1 241767682 241767683 23596 0.8125095 - 1.0680801  
## 4 chr1 243418399 243418400 9859 0.9652586 - 1.7805035  
## 5 chr1 8073372 8073373 54206 0.8477583 - 2.5678589  
## 6 chr1 8073689 8073690 54206 0.8089832 - 1.1375522

The ‘name’ column can be entrez gene ID or gene symbol.

deinfo contains the differential expresion p-value and fdr for genes. It can be made using makegrreadsfrombam and getdeinfo, or users can use any other method to make it as the following formate:

head(deinfo)

## name pval padj  
## 1 1 7.578860e-01 8.990406e-01  
## 2 100 6.958592e-01 8.695820e-01  
## 3 1000 4.155368e-06 6.420489e-05  
## 4 10000 2.043250e-02 9.424864e-02  
## 5 100009676 4.524888e-01 7.148682e-01  
## 6 10001 6.708161e-01 8.566831e-01

The ‘name’ column can be entrez gene ID or gene symbol.

bamreadsgr can be generated using makegrreadsfrombam from the MeRIP-Seq data in bam formate.

bamreadsgr <- system.file("extdata", "bamgrlist\_toy.RData", package="Funm6AViewer")  
load(bamreadsgr)  
  
siggene <- c("CCNT1", "MYC", "BCL2")  
permutime <- 1000

The datapath is the filepath where the required PPI data saved and the enrich\_input\_directory is the filepath passed to string\_db, the GO and KEGG function annotation data will be downloaded to this path. All these required data can be downloaded from <https://pan.baidu.com/s/1qOGG57OgxmrTwSbbBEeQ2w&shfl=sharepset>

datapath <- "F:/Funm6A\_package/data"  
enrich\_input\_directory <- "F:/Funm6A\_package/data"  
  
savepath <- getwd()  
re <- funm6aviewer(dminfo, deinfo, grlist, intrested\_gene = siggene, permutime = permutime,  
 datapath = datapath, enrich\_input\_directory = enrich\_input\_directory, savepath = savepath)

The results will be saved to savepath.

## 4. DmM sites plot for interested gene

Users can use dmsiteplot to visaulize the DmM sites on their interested genes and their isoforms.

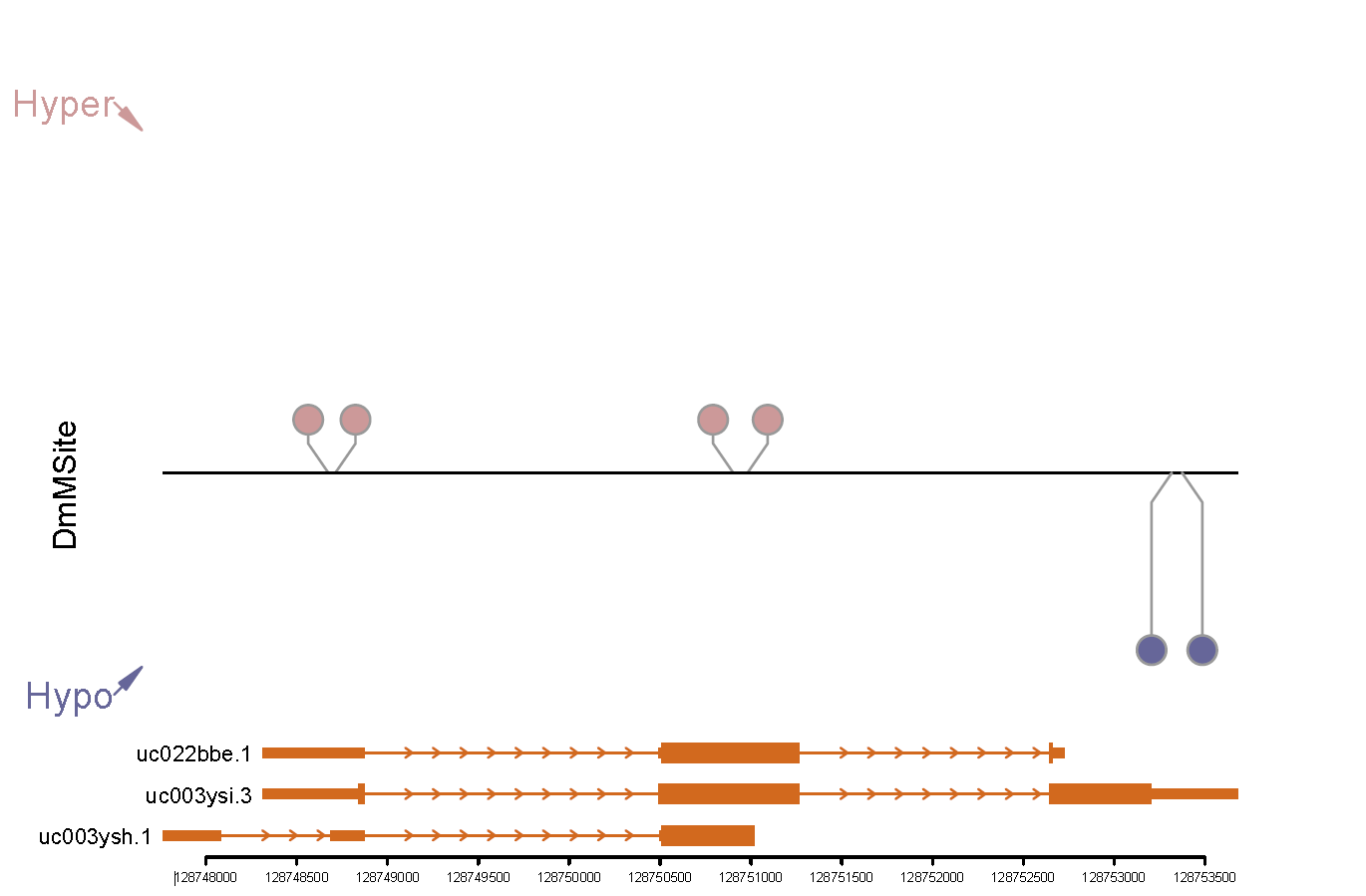
Get input:

dminfo <- system.file("extdata", "DMinfo\_toy.xls", package="Funm6AViewer")  
dminfo <- read.table(dminfo, header = TRUE, stringsAsFactors = FALSE)  
  
siggene <- c("MYC")

Make plot:

re <- dmsiteplot(dminfo = dminfo, intrested\_gene = siggene)

## [1] "total 82960 transcripts extracted ..."  
## [1] "total 63675 mRNAs left after component length filter ..."  
## [1] "total 13046 ncRNAs left after ncRNA length filter ..."



## 5. Reads coverage plot for interested gene

Users can use coverageplot to visaulize the reads coverage of DmM sites on their interested genes.

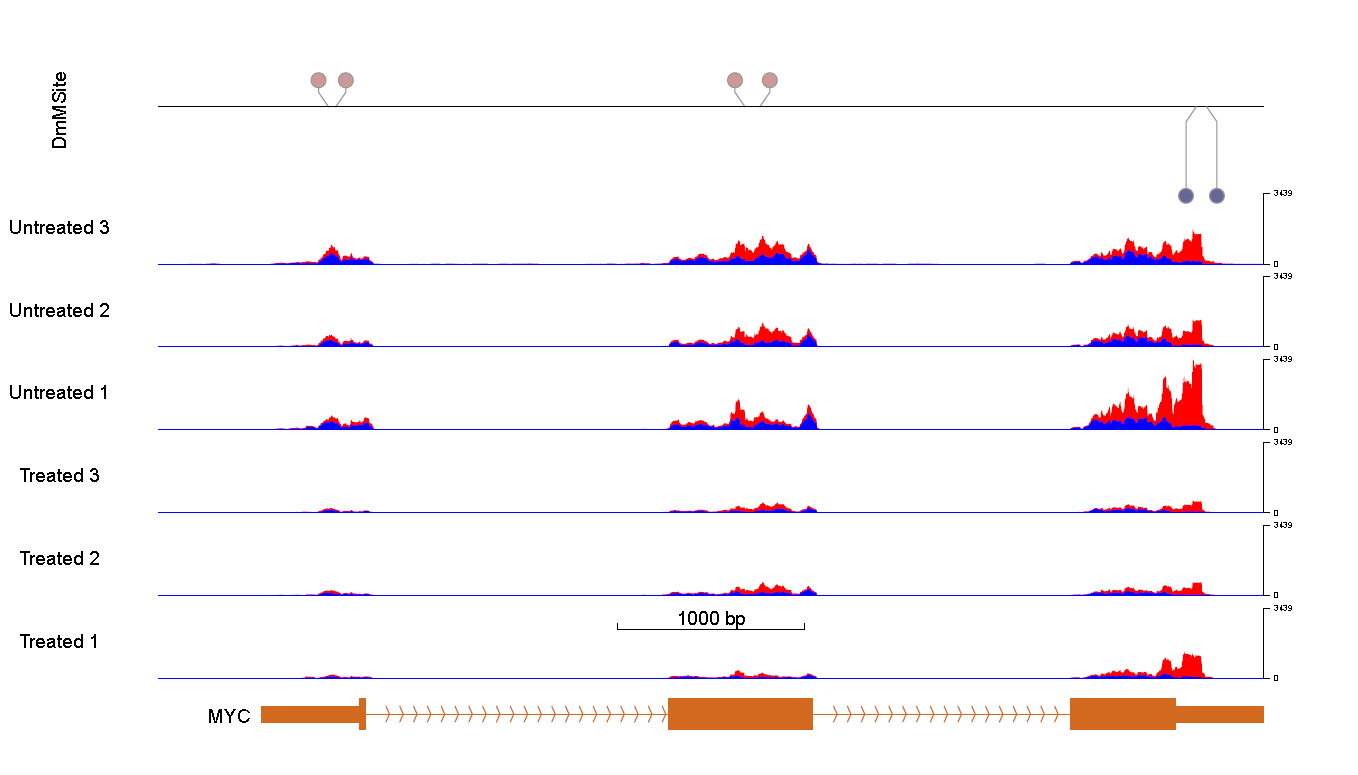
Get input:

dminfo <- system.file("extdata", "DMinfo\_toy.xls", package="Funm6AViewer")  
dminfo <- read.table(dminfo, header = TRUE, stringsAsFactors = FALSE)  
  
bamreadsgr <- system.file("extdata", "bamgrlist\_toy.RData", package="Funm6AViewer")  
load(bamreadsgr)  
  
siggene <- c("MYC")

Make plot:

re <- coverageplot(dminfo = dminfo, grlist = grlist, intrested\_gene = siggene)

## [1] "total 82960 transcripts extracted ..."  
## [1] "total 63675 mRNAs left after component length filter ..."  
## [1] "total 13046 ncRNAs left after ncRNA length filter ..."



## 6. FunDMDeep-m6A

If users only hase a list of DmM genes and their DE information, then they can use fdmdeepm6A to identify functional DmM genes (FDmMGenes).

DMgene is a group of DmM genes, can be gene symbol or entrez gene ID.

dminfo <- system.file("extdata", "DMinfo\_toy.xls", package="Funm6AViewer")  
deinfo <- system.file("extdata", "DEinfo\_toy.xls", package="Funm6AViewer")  
  
dminfo <- read.table(dminfo, header = TRUE, stringsAsFactors = FALSE)  
deinfo <- read.delim(deinfo, header = TRUE, stringsAsFactors = FALSE)  
  
DMgene <- unique(dminfo$name)  
descore <- getdescore(deinfo)

The datapath is the filepath where the required PPI data saved and it can be downloaded from <https://pan.baidu.com/s/1qOGG57OgxmrTwSbbBEeQ2w&shfl=sharepset>

datapath <- "F:/Funm6A\_package/data"  
permutime <- 1000

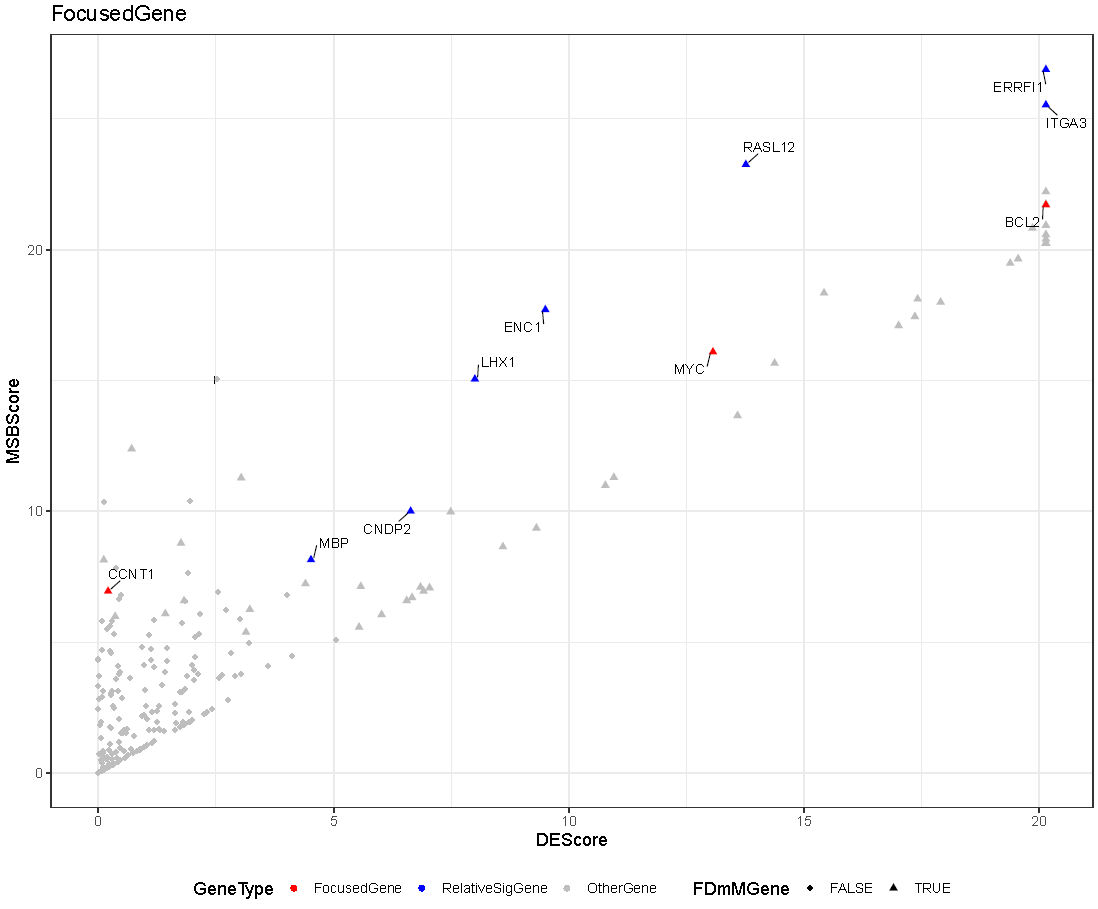
Identify FDmMGenes:

re <- fdmdeepm6A(DMgene = DMgene, descore = descore, datapath = datapath, permutime = permutime)

## [1] "hint"  
## [1] "biogrid"  
## [1] "iRef"  
## [1] "multinet"  
## [1] "1000 times random test for ranks, this may take a few hours..."

Plot interested genes’ MSB score:

siggene <- c("CCNT1", "MYC", "BCL2")  
siggenescoreplot(fdmgene = re, siggene = siggene)



## 7. Context specific function annotation of interested FDmMGenes

Users can visualize the context specific function of interested FDmMGenes identified by FunDMDeep-m6A using siggenepathplot.

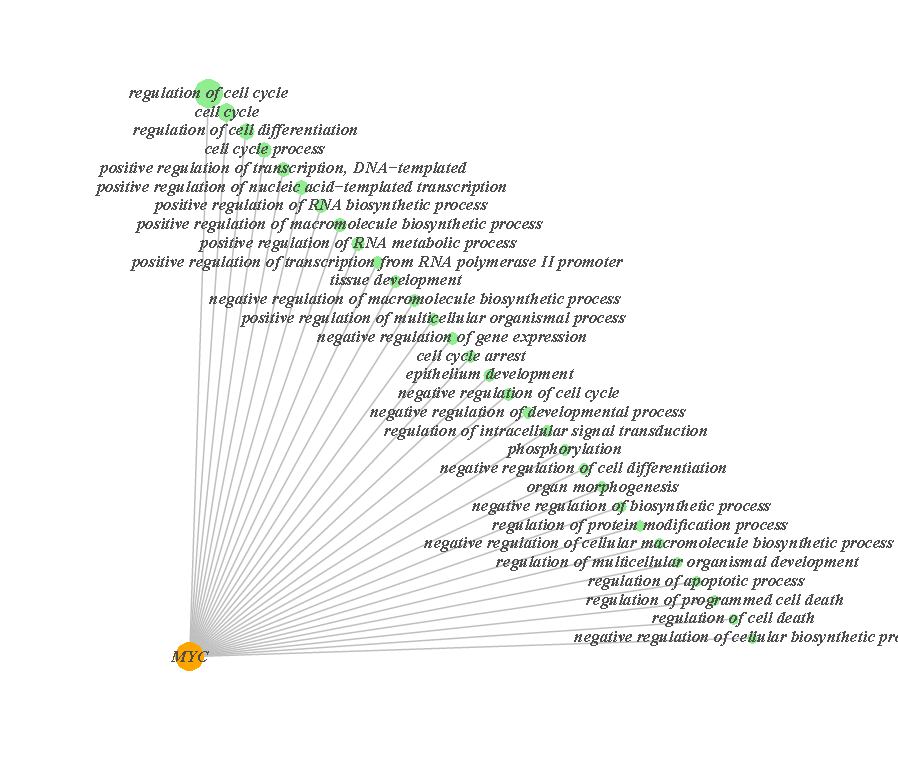
fdmgene is a group of identified FDmMGnes. siggene is interested FDmMGne.

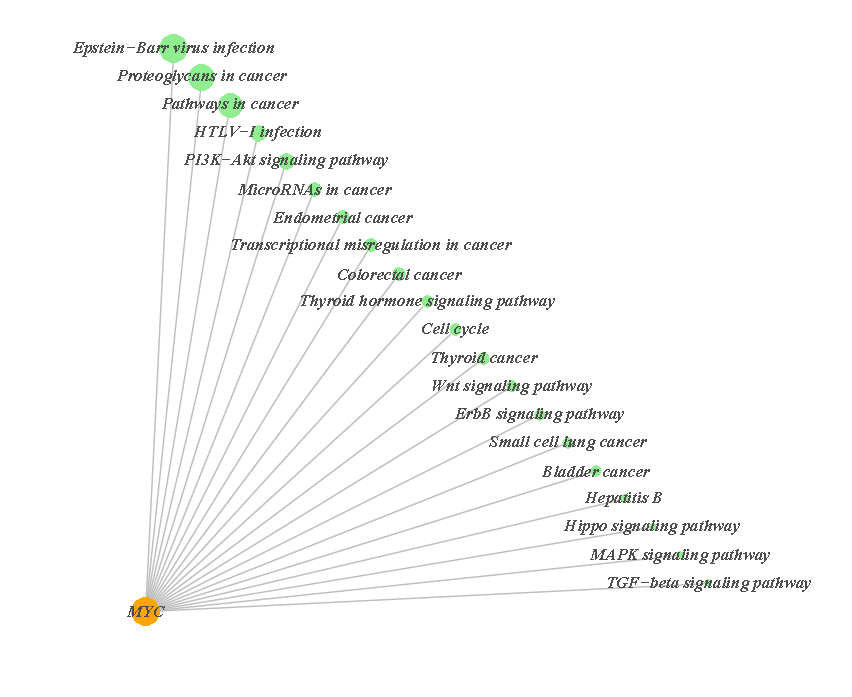
dminfo <- system.file("extdata", "DMinfo\_toy.xls", package="Funm6AViewer")  
dminfo <- read.table(dminfo, header = TRUE, stringsAsFactors = FALSE)  
fdmgene <- unique(dminfo$name)  
  
siggene <- c("MYC")

The input\_directory is the filepath passed to string\_db, the GO and KEGG function annotation data will be downloaded to this path. Users can also donwloaded the annotation data previously from <https://pan.baidu.com/s/1qOGG57OgxmrTwSbbBEeQ2w&shfl=sharepset> and set the input\_directory as where you save the data.

input\_directory <- "F:/Funm6A\_package/data"  
  
re <- siggenepathplot(fdmgene = fdmgene, intrested\_gene = siggene, input\_directory = input\_directory)

## Warning: we couldn't map to STRING 3% of your identifiers





## 8. MSB net plot for interested FDmMGenes

Users can visualize the MSB neighbours of interested FDmMGenes identified by FunDMDeep-m6A using msbnetplot.

dmgene is a group of DmM gnes. siggene is interested FDmMGnes.

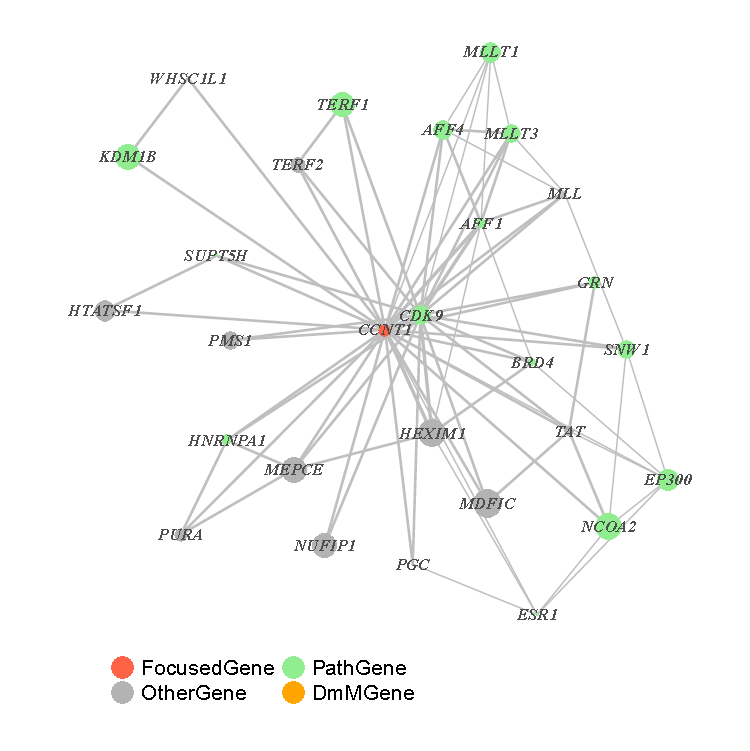
dminfo <- system.file("extdata", "DMinfo\_toy.xls", package="Funm6AViewer")  
deinfo <- system.file("extdata", "DEinfo\_toy.xls", package="Funm6AViewer")  
  
dminfo <- read.table(dminfo, header = TRUE, stringsAsFactors = FALSE)  
deinfo <- read.delim(deinfo, header = TRUE, stringsAsFactors = FALSE)  
  
dmgene <- unique(dminfo$name)  
descore <- getdescore(deinfo)  
  
siggene <- c("CCNT1", "MYC", "BCL2")

The datapath is the filepath where the required PPI data saved and it can be downloaded from <https://pan.baidu.com/s/1qOGG57OgxmrTwSbbBEeQ2w&shfl=sharepset>

datapath <- "F:/Funm6A\_package/data"

Plot for one FDmMGne:

re <- msbnetplot(genesymbol = siggene[1], dmgene = dmgene, descore = descore, datapath = datapath)



plot for several FDmMGnes:

re <- msbnetplot(genesymbol = siggene, dmgene = dmgene, descore = descore, datapath = datapath,  
 savename = "InterestedGene")

