NetLCP Tutorial

Ran MingYu

2022/5/11

Introduction

Numerous independent networks of regulatory elements including lncRNA, circRNA and pathway have been developed to crucial roles in computational system biology. Crosstalk among those networks as a bridge to construct and decode heterogenous network aids to acquire bio-meaningful information from multidimensional biological knowledge. We introduce NetLCP, an R package with command and shiny-based GUI modes, for customizing prioritization of regulatory elements in heterogenous network with low noisy and detecting mutation 'switch' in local area of network, which highlights interested regulatory elements or potential dysfunction. The latest version of NetLCP can be downloaded in https://github.com/mortyran/NetLCP. Here, we provide the whole example workflow related to cellular senescence.

Installation

```
# Method 1
library(devtools)
devtools::install_github("mortyran/NetLCP")

# Method 2
BiocManager::install('githubinstall')
library(githubinstall)
githubinstall("NetLCP")
```

Example workflow

1. Data preparation

```
cs_url = "C:\\Users\\acer\\Desktop\\NetLCP\\Paper\\Tutorial\\CellularSenescenceData"
cs_data = read.table(cs_url, sep = "\t", header = T)
head(cs_data)
```

1.1 First, we read in the cellular senescence gene set as example transcriptome from github.

```
  Symbol                Type Trans_ID
1   AKT1 cellular senescence      207
2   ATF2 cellular senescence     1386
3   BDNF cellular senescence      627
```

```
4    CDK1 cellular senescence      983
5 CDKN2A cellular senescence      1029
6     CLU cellular senescence      1191
```

1.2 Initialize NetLCP.

1.2.1 (Recommended) Manually download data from internet depository and initialize NetLCP. After acquiring the file named "NetLCPData.tar.gz" from https://www.dropbox.com/s/4dqtxecpw79iejx/NetLCPData.tar.gz or http://hainmu-biobigdata.com/NetLCP/NetLCPData.tar.gz, please following steps:

Step1: Get the NetLCP package preset data directory.

```
NetLCP_file_path = system.file("extdata", package = "NetLCP")
```

Step2: Copy the downloaded file "NetLCPData.tar.gz" to the directory.

```
# your_file_path is the file path of "NetLCPData.tar.gz" in your computer.
file.copy(your_file_path, NetLCP_file_path)
```

Step3: Run dataInitialize() to initialize the data in R.

```
library(NetLCP)
```

```
NetLCP,
More information can be found at https://github.com/rmyhandsome/NetLCP
If you use NetLCP in you publication, please cite this publication:
NetLCP: An R package for customizing prioritization of regulatory elements in heterogeneous network with
Authors: MingYu Ran (rmyhandsome@163.com)
Maintainer: MingYu Ran.
Please read the tutorial in https://mortyran.github.io/NetLCP/ for data preparation before you use NetLC
```

```
dataInitialize()
```

```
[1] "Data initialization has been finished!"
```

```
library(NetLCP)
dataInitialize()
```

1.2.2 Automatically initialize NetLCP, this will depend on your network connection.

2. Prioritize the biological elements.

Prioritize the biological elements in heterogenous network by input transcriptome (mRNA/miRNA, but mixed miRNA and mRNA is highly recommended). NetLCP supports miRBase ID for miRNA and Entrez ID for mRNA. The parameter **transcriptomeList** is the input transcriptome. The **prioType** represents the biological element type, which currently contains lncRNA, circRNA and pathway(KEGG, Reactome and Wikipathway). **empiricalPvalue** is alternative for an empirical p.value through random disturbance, default value is FALSE(it could take several hours).

```
lncRNA_prio = BioRegElePrioritization( transcriptomeList = cs_data$Trans_ID,
                                        prioType = "lncRNA",
                                        empiricalPvalue = FALSE )
```

2.1 lncRNA prioritization

```
[1] "Filtering the missing elements of transcriptomeList in the input network......"
[1] "Element 10934/7012 have been filtered....."
[1] "Now remain 367"
[1] "Prioritization begins, please wait while we do something......"
[1] "Prioritization finished......"
```

```
head(lncRNA_prio)
```

```
        NodeName Ranking OfficialName NorRiskScore
1 ENSG00000229807       1         XIST     5.808779
2 ENSG00000247556       2      OIP5-AS1     5.377587
3 ENSG00000251562       3       MALAT1     5.117147
4 ENSG00000225733       4      FGD5-AS1     4.810600
5 ENSG00000230551       5         <NA>     4.662645
6 ENSG00000231074       6        HCG18     4.474082
```

```
circRNA_prio = BioRegElePrioritization( transcriptomeList = cs_data$Trans_ID,
                                         prioType = "circRNA",
                                         empiricalPvalue = FALSE )
head(circRNA_prio)
```

2.2 circRNA prioritization

```
KEGG_prio = BioRegElePrioritization( transcriptomeList = cs_data$Trans_ID,
                                      prioType = "KEGG",
                                      empiricalPvalue = FALSE )
```

2.3 KEGG pathway prioritization

```
[1] "Filtering the missing elements of transcriptomeList in the input network......"
[1] "Element 10934/7012 have been filtered....."
[1] "Now remain 367"
[1] "Prioritization begins, please wait while we do something......"
[1] "Prioritization finished......"
```

```
head(KEGG_prio)
```

```
  NodeName Ranking                                        OfficialName NorRiskScore
1 hsa05200       1                                   Pathways in cancer     2.541762
2 hsa05206       2                                   MicroRNAs in cancer     2.450967
3 hsa05167       3 Kaposi sarcoma-associated herpesvirus infection     2.359666
4 hsa04218       4                                 Cellular senescence     2.180467
5 hsa05165       5                        Human papillomavirus infection     2.115676
6 hsa05203       6                                Viral carcinogenesis     2.043223
```

```
Reactome_prio = BioRegElePrioritization( transcriptomeList = cs_data$Trans_ID,
                                          prioType = "Reactome",
                                          empiricalPvalue = FALSE )
head(Reactome_prio)
```

2.4 Reactome pathway prioritization

```
Wikipathway_prio = BioRegElePrioritization( transcriptomeList = cs_data$Trans_ID,
                                            prioType = "Wikipathway",
                                            empiricalPvalue = FALSE )
head(Wikipathway_prio)
```

2.5 Wikipathway pathway prioritization

3. Inspect local area of heterogenous network

Current regulation types in local area of heterogenous network include binary elements regulation and multiple elements regulation. NetLCP will map the experimentally verified interactions between input biological elements to the local area of heterogenous network. NetLCP supports Ensembl ID for lncRNA, miRBase ID for miRNA, Entrez ID for mRNA, KEGG ID, Reactome ID, Wikipathway ID for pathway.

Here we explore the regulation among the top 10 of lncRNA and KEGG prioritization results and example transcriptome in local area.

```
# Input data preparation
lncRNA_top10 = lncRNA_prio$NodeName[1:10]
KEGG_top10 = KEGG_prio$NodeName[1:10]
local_data = c(lncRNA_top10, KEGG_top10, cs_data$Trans_ID)
```

3.1 Binary elements regulation   Inspect the binary elements regulation in the local area of heterogenous network. **transcriptomeList** represents the biological elements in the interested local area of heterogenous network. **regulationType** represents the regulation type in local area including circRNA-miRNA, lncRNA-miRNA, lncRNA-mRNA, miRNA-mRNA, miRNA-pathway, mRNA-pathway. **allRegulation** is a logical value. If you set "FALSE", it will return the regulation in the local area of heterogenous network which only contains biological elements in the input transcriptomeList, i.e. local_data. If you set "TRUE", it will search the regulation in the whole depository, which means the results can contain other biological elements. In this case, you can regarded as NetLCP as an independent depository to only extract associated regulatory data. "FALSE" (default) is a common setting.

```
# miRNA-mRNA regulation in local area
bi_local_miRNA_mRNA = binaryRegulation( transcriptomeList = local_data,
                                        regulationType = "miRNA-mRNA",
                                        allRegulation = FALSE )
# miRNA-pathway regulation in local area
bi_local_miRNA_pathway = binaryRegulation( transcriptomeList = local_data,
                                           regulationType = "miRNA-pathway",
                                           allRegulation = FALSE )
```

3.2 Multiple elements regulation Inspect the multiple elements regulation in the local area of heterogenous network.**regulationType** represents the regulation type in local area including lncRNA-miRNA-mRNA, circRNA-miRNA-mRNA, lncRNA-miRNA-mRNA-pathway and circRNA-miRNA-mRNA-pathway. Other parameters are the same as **binaryRegulation** function. Here we concentrate on the lncRNA-miRNA-mRNA-pathway regulation in the local area.

```
multi_local = multieleRegulation( transcriptomeList = local_data,
                                  regulationType = "lncRNA-miRNA-mRNA-pathway",
                                  allRegulation = FALSE )
```

```
[1] "Filtering the missing input transcriptome in input network......"
[1] "10934/7012 have been filtered....."
[1] "Now remain 387"
[1] "Multielement regulation extraction begins, please wait while we do something......"
```

```
head(multi_local)
```

```
         node1           node2                                  source
1: MIMAT0000062 ENSG00000255717              LncBaseExperimental
2: MIMAT0000449 ENSG00000251562         LncBaseExperimental;RAIN
3: MIMAT0000691 ENSG00000231074                           NPInter
4: MIMAT0000691 ENSG00000270604                           NPInter
5: MIMAT0000681 ENSG00000269821     LncBaseExperimental;NPInter;RAIN
6: MIMAT0000245 ENSG00000231074 LncBaseExperimental;lncRNASNP2;ENCORI
        regType
1: lncRNA-miRNA
2: lncRNA-miRNA
3: lncRNA-miRNA
4: lncRNA-miRNA
5: lncRNA-miRNA
6: lncRNA-miRNA
```

3.3 Biological elements statistics You can calculate the degree of biological elements in the local heterogenous network and customize the network visualization of results.

regData is the standard output of **binaryRegulation** or **multieleRegulation** functions, **filterDegree** means filtering the nodes which are less than it. **selectNode** represents certain or a group of elements you specify. **netLayout** is the alternative layout of network, "layout_nicely" or "layout_in_circle" (If the network is too large to exhibit, try to use this layout).

```
# Calculate the degree of biological elements in the local heterogenous network.
regStat( regData = multi_local,
         filterDegree = 40,
         selectNode = NULL )
```
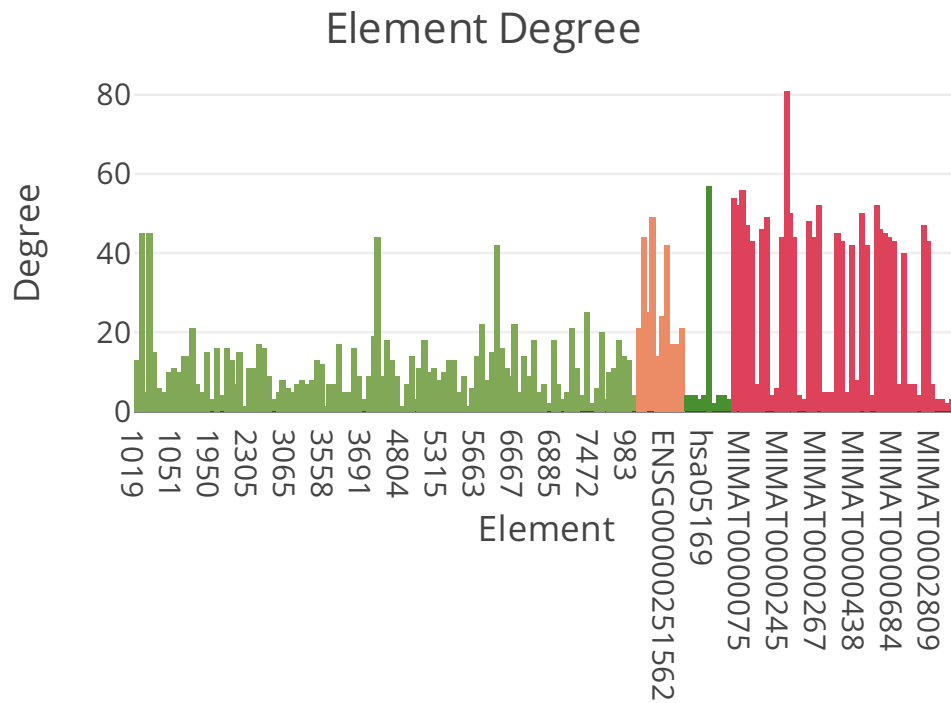
Element Degree
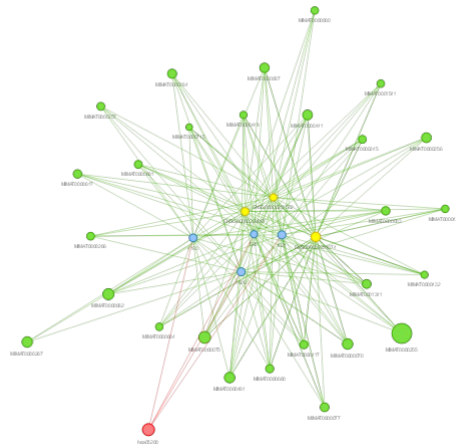


```
# network visualization.
regNetVis( regData = multi_local,
           filterDegree = 40,
           selectNode = NULL,
           netLayout = "layout_nicely" )
```

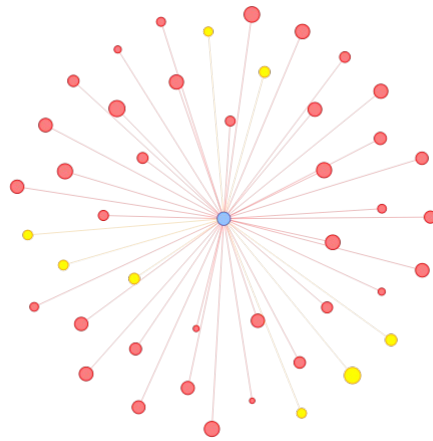Select by id ▼

Select by group ▼

🖊 Edit


```
# if you want to see the associated regulation of the biological
# element "4193" in local area of heterogeous network.
regNetVis( regData = multi_local,
           filterDegree = 40,
           selectNode = "4193",
           netLayout = "layout_nicely" )
```

4. Detecting mutation 'switch' on local area of heterogenous network.

Mutation 'switch' includes eQTLs of single biological element and the regulatory mutations on its binding site.

4.1 eQTLs of single biological elements and relative statistics.

4.1.1 Detect the eQTLs in the local area of heterogenous network. **regData** is the standard output of **binaryRegulation** or **multieleRegulation** functions.

```
eQTLsData = eQTLsDetection(regData = multi_local)
```

```
[1] "Single biological elements eQTLs extracting extracting begins......"
```

```
head(eQTLsData)
```

```
    Elements       SNPID       Location Source      RegType DataSource
1:       207 rs149007767  chr7:50370254  blood trans-eQTLs     eQTLGen
2:       595  rs67250450  chr7:28174986  blood trans-eQTLs     eQTLGen
3:       595   rs4722762  chr7:28178262  blood trans-eQTLs     eQTLGen
4:       595    rs917116  chr7:28172739  blood trans-eQTLs     eQTLGen
5:       604   rs7210990 chr17:16170764  blood trans-eQTLs     eQTLGen
6:      1026  rs12589699 chr14:69212885  blood trans-eQTLs     eQTLGen
```

4.1.2 eQTLs statistics You can perform statistics on the eQTLs of single elements or regulation and customize the network visualization of results. **regData** is the standard output of **binaryRegulation** or **multieleRegulation** functions, **eQTLsData** is the standard output of "eQTLsDetection". **regulationType** represent the regulation type of **regData**, which supports circRNA-miRNA, lncRNA-miRNA, lncRNA-mRNA, miRNA-mRNA, miRNA-pathway, mRNA-pathway, circRNA-miRNA-mRNA, lncRNA-miRNA-mRNA, miRNA-mRNA-pathway, lncRNA-miRNA-mRNA-pathway and circRNA-miRNA-mRNA-pathway. **filterDegree** means filtering the nodes which are less than it. **selectNode** represents a group of elements you specify. **netLayout** is the alternative layout of network, "layout_nicely" or "layout_in_circle" (If the network is too large to exhibit, try to use this layout).

```
# count the eQTLs of single elements.
eQTLsSingleEleStat( regData = multi_local,
                    eQTLsData = eQTLsData,
                    filterDegree = 50,
                    selectNode = NULL )
```
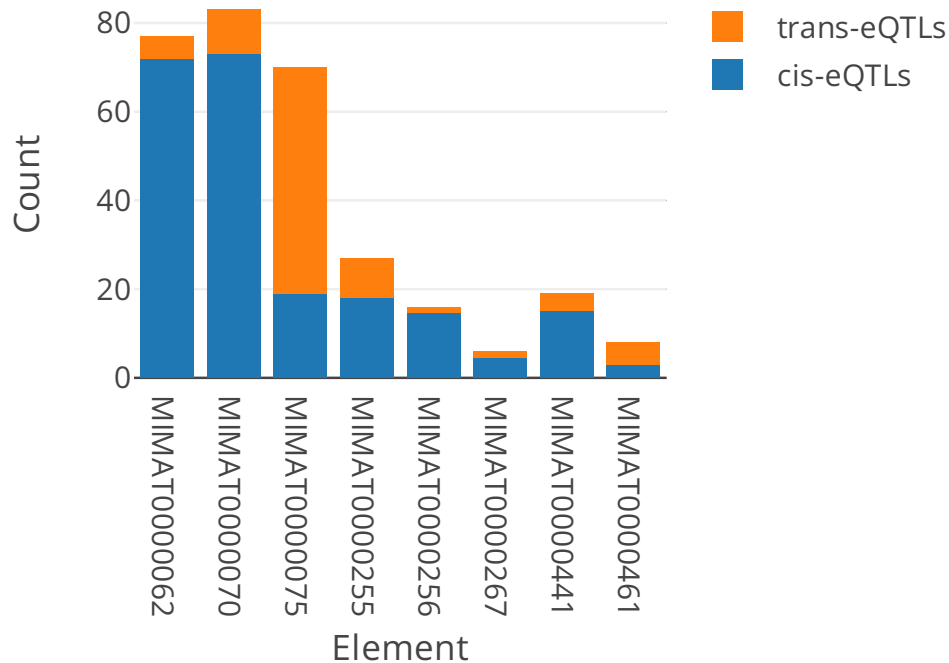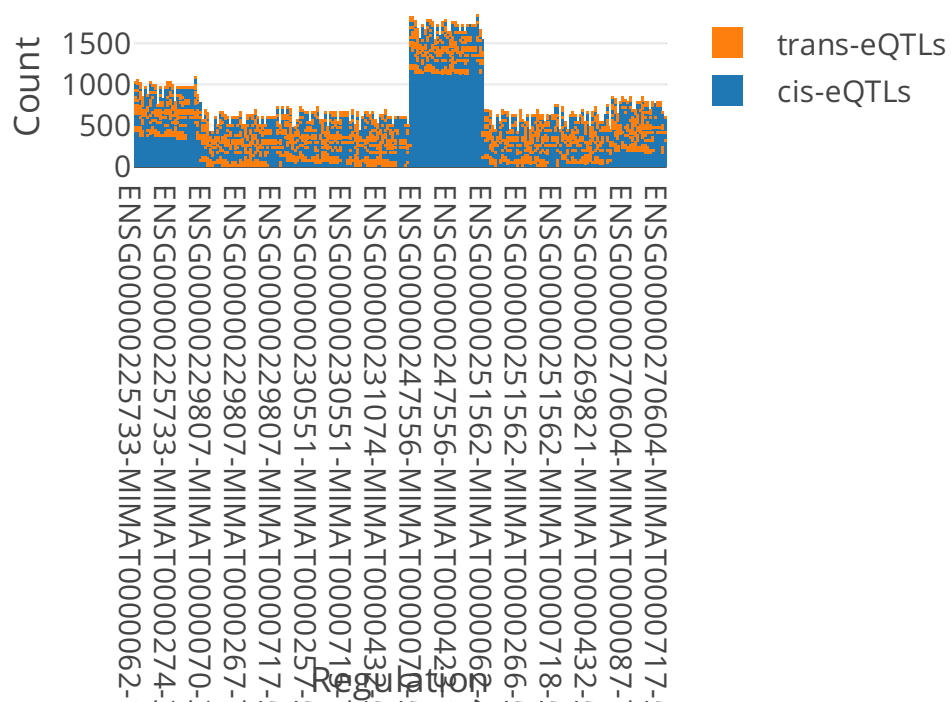


```
# count the eQTLs of multiple elements on regulation.
eQTLsRegStat( regData = multi_local,
              eQTLsData = eQTLsData,
              regulationType = "lncRNA-miRNA-mRNA-pathway",
```

```
          filterDegree = 30,
          selectNode = NULL )
```
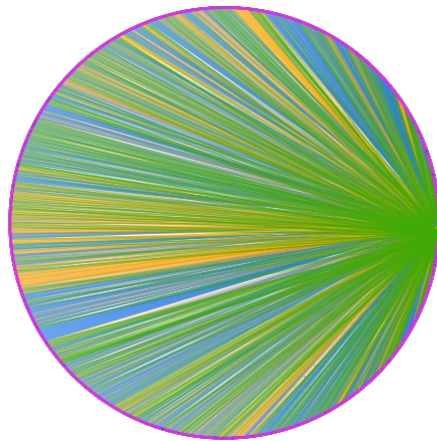


```
# network visualization.
eQTLsNetVis( regData = multi_local,
             eQTLsData = eQTLsData,
             filterDegree = 30,
             selectNode = NULL,
             netLayout = "layout_in_circle" )
```

```
# Give that you highlight the regulation "ENSG00000251562 - MIMAT0000081 - 4193 - hsa04218"
# by statistics, you want to concentrate on the eQTLs of the biological elements in this regulation.
eQTLsNetVis(regData = multi_local,
            eQTLsData = eQTLsData,
            filterDegree = 30, # this parameter will be automatically ignored when setting "selectNode"
            selectNode = c("ENSG00000251562", "MIMAT0000081", "4193", "hsa04218"),
            netLayout = "layout_nicely")
```
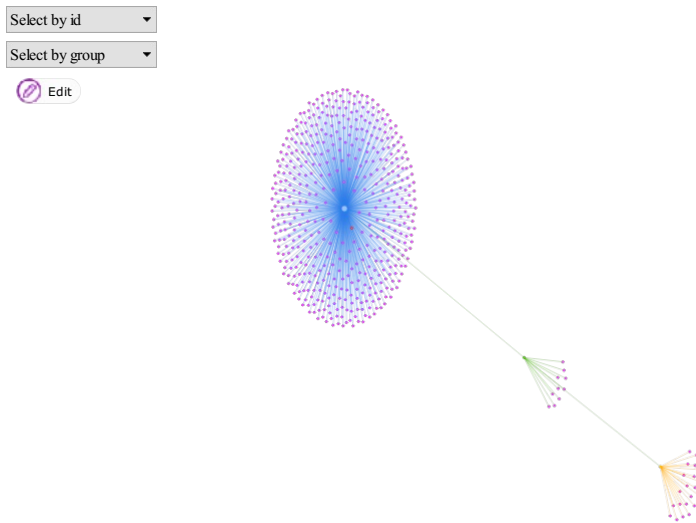
4.2 Regulatory mutations on the binding site of biological elements

4.2.1 Detect the regulatory mutations on the binding site of biological elements in the local area of heterogenous network **regData** is the standard output of **binaryRegulation** or **multieleRegulation** functions. **regulationType** represent the regulation type of **regData**, which supports "miRNA-mRNA", "miRNA-mRNA-pathway", "lncRNA-miRNA-mRNA", "circRNA-miRNA-mRNA", "lncRNA-miRNA-mRNA-pathway" or "circRNA-miRNA-mRNA-pathway".

```
regMutData = regVarDetection(regData = multi_local, regulationType = "lncRNA-miRNA-mRNA-pathway")
```

```
[1] "Variants on regulations extracting begins......"
```

```
head(regMutData)
```

```
            ceRNA          miRNA  mRNA Location Population lncMutType Source1
1: ENSG00000231074 MIMAT0000441 23411   6p22.1        UVM        CNV LnCeVar
2: ENSG00000231074 MIMAT0000441 23411   6p22.1        UVM        CNV LnCeVar
3: ENSG00000231074 MIMAT0000441 23411   6p22.1        UVM        CNV LnCeVar
4: ENSG00000231074 MIMAT0000441 23411   6p22.1        UVM        CNV LnCeVar
5: ENSG00000231074 MIMAT0000441 23411   6p22.1        UVM        CNV LnCeVar
6: ENSG00000231074 MIMAT0000441 23411   6p22.1        UVM        CNV LnCeVar
                 mim       miRNAID geneID       SNPID mimMutType    Source2
1: MIMAT0000441-23411 MIMAT0000441  23411 rs971258276   UTR3_SNP miRNASNPv3
2: MIMAT0000441-23411 MIMAT0000441  23411 rs971258276   UTR3_SNP miRNASNPv3
3: MIMAT0000441-23411 MIMAT0000441  23411 rs200484021   UTR3_SNP miRNASNPv3
4: MIMAT0000441-23411 MIMAT0000441  23411 rs200484021   UTR3_SNP miRNASNPv3
```

```
5: MIMAT0000441-23411 MIMAT0000441  23411 rs201624047   UTR3_SNP miRNASNPv3
6: MIMAT0000441-23411 MIMAT0000441  23411 rs201624047   UTR3_SNP miRNASNPv3
    pathway pathwaySource      regType
1: hsa04218          KEGG mRNA-pathway
2: hsa05206          KEGG mRNA-pathway
3: hsa04218          KEGG mRNA-pathway
4: hsa05206          KEGG mRNA-pathway
5: hsa04218          KEGG mRNA-pathway
6: hsa05206          KEGG mRNA-pathway
```

4.2.2 Regulatory mutations statistics "regMutData" is the standard output of **regVarDetection** function. **regulationType** is the same as parameter in **regVarDetection** function. **selectNode** is always needed and only accept a group of elements in regulation.

```
# count the regulatory mutations on the binding sites of biological elements in regulation.
regVarStat(regVar = regMutData,
          regulationType = "lncRNA-miRNA-mRNA-pathway",
          selectNode = c("ENSG00000251562", "MIMAT0000081", "4193", "hsa04218", "ENSG00000247556"))
```
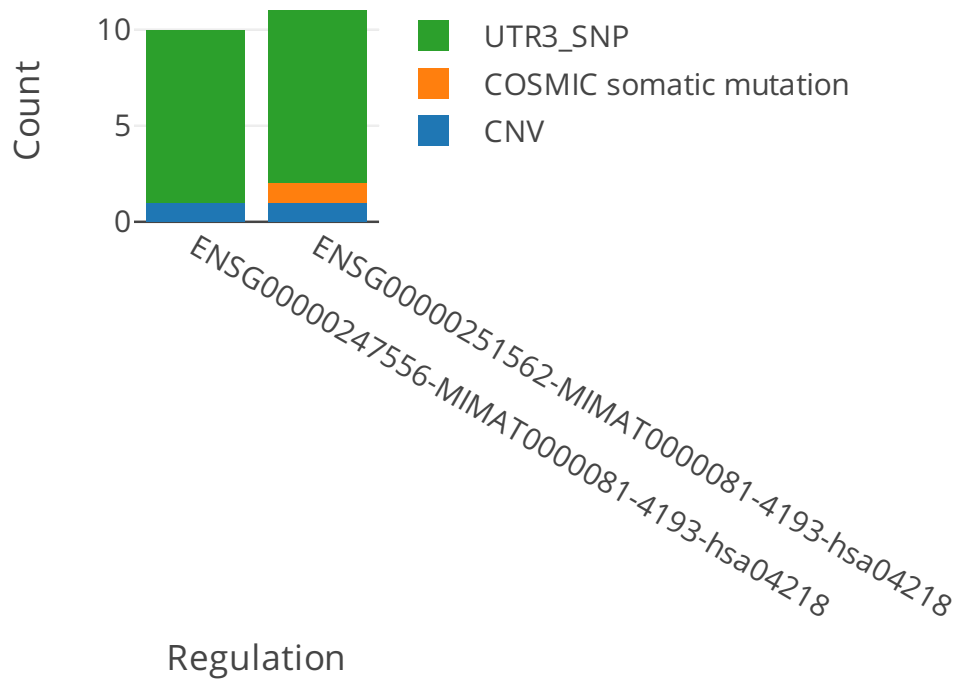
```
# network visualization.
regVarNetVis(regVar = regMutData,
             regulationType = "lncRNA-miRNA-mRNA-pathway",
             selectNode = c("ENSG00000251562", "MIMAT0000081", "4193", "hsa04218"))
```
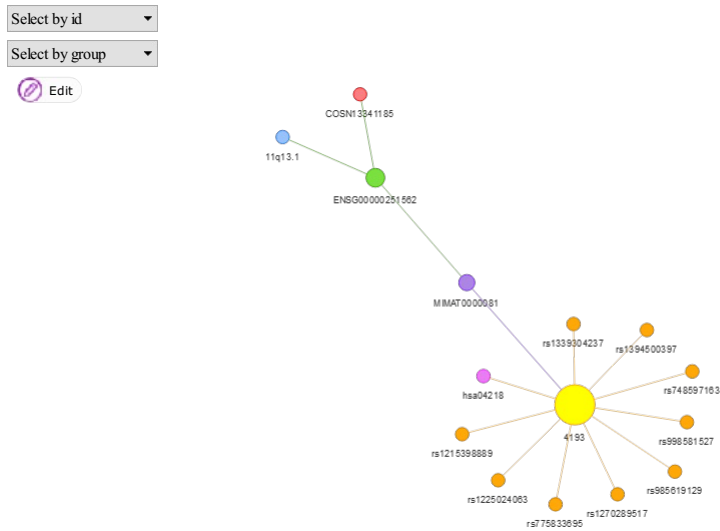

If you have any questions, please contact us without hesitation.

Ming-Yu, Ran

Email: rmyhandsome@163.com