Supplementary File 1

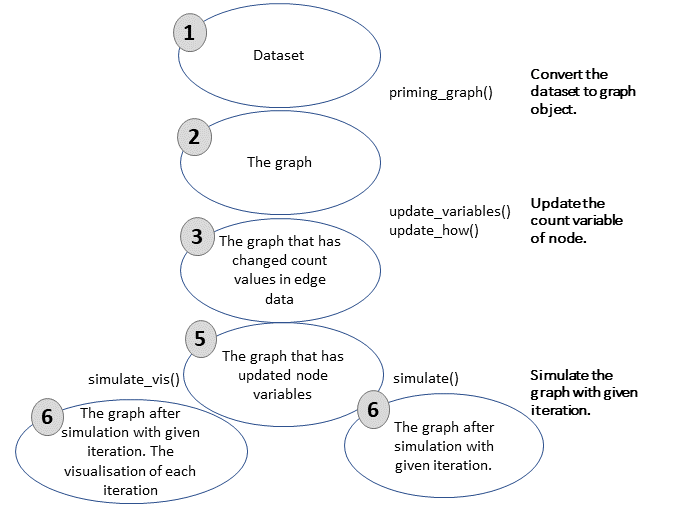
Selcen Ari

7 Nisan 2019

# 1. Defined functions for ceRNA models and workflow of method

We defined the functions that can be used with R programming. Briefly, these functions process a given miRNA:gene dataset and convert to graph object. All values that are significant in miRNA:target interactions are stored in edge variables and processed with formulations that are given in previous section. The functions and steps of approach are explained as following (Figure S1) :

**Convertion of dataset**: priming\_graph function processes the given dataset that includes competing elements in first variable and repressive element in second variable. If the affinity and/or degradation factors are specified in the function, factors are taken into account, are processed with defaults in vice versa. The formulations that are given in equations (1-4) are performed in this function. This step gives the graph object which contains efficiency values of miRNA:competing target pairs in steady-state in terms of amount. It is assumed that the initial target amounts in the dataset is observed after the reppressive activity of miRNAs in steady-state.



Workflow for simulation of competing endogenous RNA regulations. Graph object in steps 2-6 is saved and updated continuously.

**Transition of variables in graph**: In the previous step, the calculations are performed in the edge variables of the graph object. However, the graph object allows to use node variables, while the node features are handled to the graph. In this direction, update\_nodes function carries the amount values to node variables This step must be applied with “once” option because it is primary process.

**Trigger change in graph**: The dataset are assumed as steady-state in previous step and the efficieny coefficients are calculated according to this acceptance. In the network that is found in steady-state conditions, the change is applied to the graph object for disturbtion of steady-state. To provide the disturbtion in the network the workflow offer two methods: update\_variables and update\_how. The first, a new dataset that is contained competing and repressive element names and current values of these can be processed with update\_variables. The second option, the amount of the given node name in update\_how function can be changed according to “how” argument.

**Updating current values of variables**: After variables updating in edge varibles, these are carried to node variables. Current and previous values of variables are stored as node variables with update\_variables function.

**Simulation of competing behavior of targets**: After the change in the steady-state conditions, the network elements try to gain steady-state again. This process progresses as repeating of regulations after the spreading the changes in the network. In this step, simulation of regulations according to given cycle count in simulate function is applied. After each simulation cycle, the miRNA repression values are re-calculated and the current values of competing elements are found and saved. The process is performed in the edge data and at the same time outputs of the calculations are carried from edge to node data.

The node elements in the dataset are handled as two type; repressives (miRNAs) and competings (targets). It is assumed in approach that while targets are degrading or inhibiting by miRNAs continuously, miRNAs reversibly used. If the trigger of the network is a miRNA, it maintains the current value of amount that provides by user. On the contrary, it tries to help this process to provide steady-state through the regulations on its amount, if a competing element is used as a trigger. The functions that are used in the approach are developed with R programming so as can be used with other packages. These are can be found in the github repository [ceRNAnetsim github page](https://github.com/selcenari/ceRNAnetsim) and improved with contributions of others.

#install.packages("devtools")  
#devtools::install\_github("selcenari/ceRNAnetsim")  
library(ceRNAnetsim)

* load *minsamp* data

data("minsamp")  
  
minsamp

## competing miRNA Competing\_expression miRNA\_expression seed\_type region  
## 1 Gene1 Mir1 10000 1000 0.43 0.30  
## 2 Gene2 Mir1 10000 1000 0.43 0.01  
## 3 Gene3 Mir1 5000 1000 0.32 0.40  
## 4 Gene4 Mir1 10000 1000 0.23 0.50  
## 5 Gene4 Mir2 10000 2000 0.35 0.90  
## 6 Gene5 Mir2 5000 2000 0.05 0.40  
## 7 Gene6 Mir2 10000 2000 0.01 0.80  
## energy  
## 1 -20  
## 2 -15  
## 3 -14  
## 4 -10  
## 5 -12  
## 6 -11  
## 7 -25

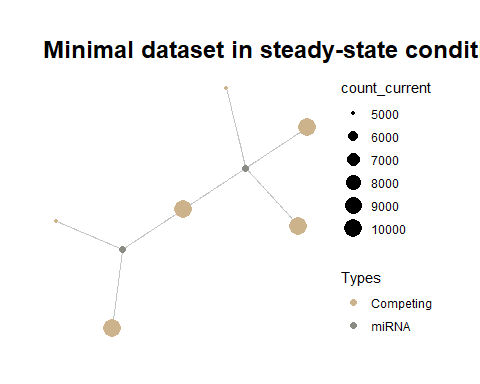
See Figure S1 in Supplementary Tables file.

## *minsamp* dataset analysis in lack of interaction factors.

Firstly, we have analysed minimal data without interaction factors between miRNA:target.

* 1. We have evaluated graph in the steady state conditions as followings:

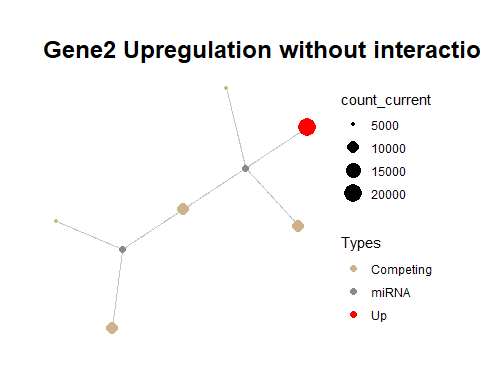
priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression)%>%  
 vis\_graph(Competing\_color = "navajowhite3", mirna\_color = "ivory4", title = "Minimal dataset in steady-state conditions")



Minimal Dataset in Steady-state

* 1. We have obtained graph after change on Gene2 expression as followings:

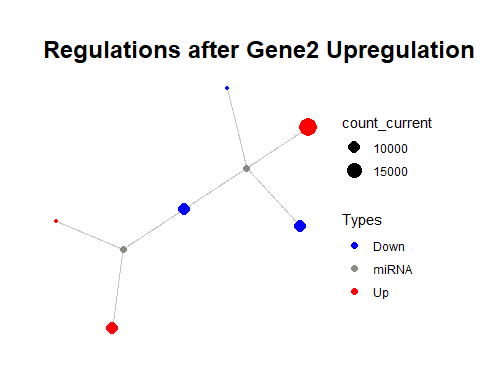
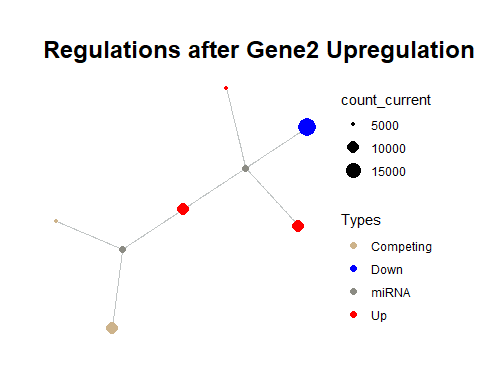
priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression)%>%  
 update\_how("Gene2", 2)%>%  
 vis\_graph(Competing\_color = "navajowhite3", mirna\_color = "ivory4", Upregulation = "red", title = "Gene2 Upregulation without interaction factors")



Gene2 Upregulation on Minimal Dataset

* 1. We have determined regulations after Gene2 Upregulation:

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression)%>%  
 update\_how("Gene2", 2)%>%  
 simulate\_vis(Competing\_color = "navajowhite3", mirna\_color = "ivory4", Upregulation = "red", Downregulation = "blue", title = "Regulations after Gene2 Upregulation", 2)



## # A tbl\_graph: 8 nodes and 7 edges  
## #  
## # A rooted tree  
## #  
## # Node Data: 8 x 7 (active)  
## name type node\_id initial\_count count\_pre count\_current  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Gene1 Comp~ 1 10000 10063. 10062.  
## 2 Gene2 Comp~ 2 10000 19841. 19845.  
## 3 Gene3 Comp~ 3 5000 5032. 5031.  
## 4 Gene4 Comp~ 4 10000 10063. 10059.  
## 5 Gene5 Comp~ 5 5000 5000 5001.  
## 6 Gene6 Comp~ 6 10000 10000 10002.  
## # ... with 2 more rows, and 1 more variable: changes\_variable <chr>  
## #  
## # Edge Data: 7 x 20  
## from to Competing\_name miRNA\_name Competing\_expre~ miRNA\_expression  
## <int> <int> <chr> <chr> <dbl> <dbl>  
## 1 1 7 Gene1 Mir1 10000 1000  
## 2 2 7 Gene2 Mir1 10000 1000  
## 3 3 7 Gene3 Mir1 5000 1000  
## # ... with 4 more rows, and 14 more variables: dummy <dbl>,  
## # afff\_factor <dbl>, degg\_factor <dbl>, comp\_count\_list <list>,  
## # comp\_count\_pre <dbl>, comp\_count\_current <dbl>,  
## # mirna\_count\_list <list>, mirna\_count\_pre <dbl>,  
## # mirna\_count\_current <dbl>, mirna\_count\_per\_dep <dbl>,  
## # effect\_current <dbl>, effect\_pre <dbl>, effect\_list <list>,  
## # mirna\_count\_per\_comp <dbl>

Note that the regulations are colored according to expression changes of present and a previous value. So, it can be observed that whole gene expressions increase in comparison of initial steady-state. The overall regulations of gene expressions are as followings:

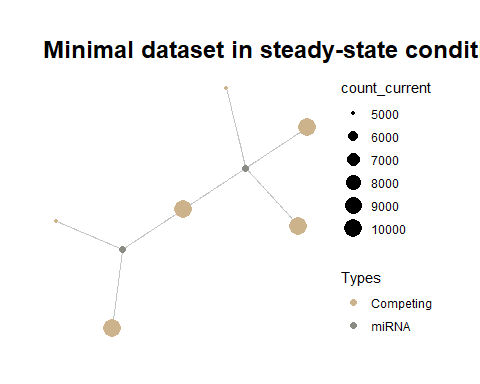
priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression)%>%  
 update\_how("Gene2", 2)%>%  
 simulate(2)%>%  
 activate(edges)%>%  
 as\_tibble()%>%  
 select(Competing\_name,comp\_count\_list,effect\_list)%>%  
 unnest()

## # A tibble: 21 x 3  
## Competing\_name comp\_count\_list effect\_list  
## <chr> <dbl> <dbl>  
## 1 Gene1 10000 286.  
## 2 Gene1 10063. 222.  
## 3 Gene1 10062. 224.  
## 4 Gene2 10000 286.  
## 5 Gene2 19841. 444.  
## 6 Gene2 19845. 441.  
## 7 Gene3 5000 143.  
## 8 Gene3 5032. 111.  
## 9 Gene3 5031. 112.  
## 10 Gene4 10000 286.  
## # ... with 11 more rows

## *minsamp* dataset analysis with interaction factors.

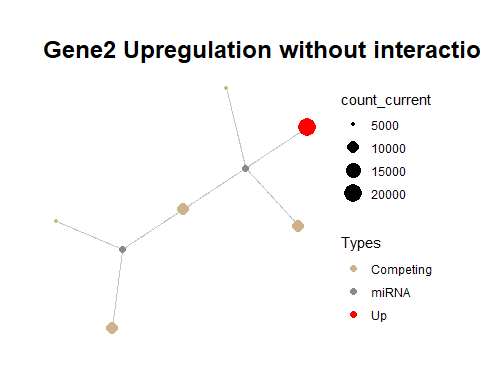
We have made the same analysis in present of interaction factors.

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)%>%  
 vis\_graph(Competing\_color = "navajowhite3", mirna\_color = "ivory4", title = "Minimal dataset in steady-state conditions")



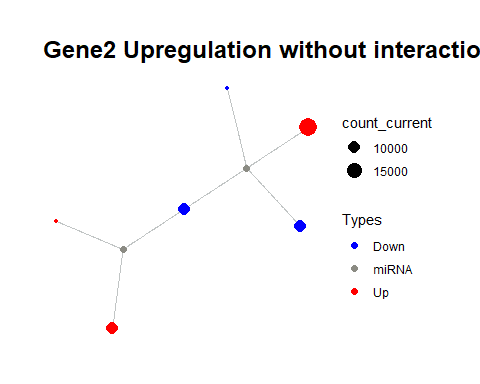
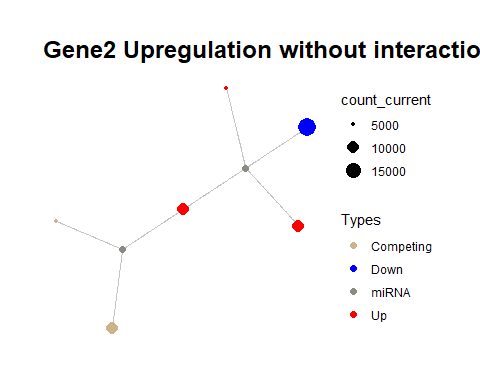
Minimal Dataset with interaction factors in Steady-state

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)%>%  
 update\_how("Gene2", 2)%>%  
 vis\_graph(Competing\_color = "navajowhite3", mirna\_color = "ivory4", Upregulation = "red", title = "Gene2 Upregulation without interaction factors")



Gene2 Upregulation on Minimal Dataset with interaction factors

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)%>%  
 update\_how("Gene2", 2)%>%  
 simulate\_vis(Competing\_color = "navajowhite3", mirna\_color = "ivory4", Upregulation = "red", title = "Gene2 Upregulation without interaction factors",2)



## # A tbl\_graph: 8 nodes and 7 edges  
## #  
## # A rooted tree  
## #  
## # Node Data: 8 x 7 (active)  
## name type node\_id initial\_count count\_pre count\_current  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Gene1 Comp~ 1 10000 10065. 10064.  
## 2 Gene2 Comp~ 2 10000 19997. 19997.  
## 3 Gene3 Comp~ 3 5000 5023. 5023.  
## 4 Gene4 Comp~ 4 10000 10029. 10028.  
## 5 Gene5 Comp~ 5 5000 5000 5000.  
## 6 Gene6 Comp~ 6 10000 10000 10000.  
## # ... with 2 more rows, and 1 more variable: changes\_variable <chr>  
## #  
## # Edge Data: 7 x 23  
## from to Competing\_name miRNA\_name Competing\_expre~ miRNA\_expression  
## <int> <int> <chr> <chr> <dbl> <dbl>  
## 1 1 7 Gene1 Mir1 10000 1000  
## 2 2 7 Gene2 Mir1 10000 1000  
## 3 3 7 Gene3 Mir1 5000 1000  
## # ... with 4 more rows, and 17 more variables: energy <dbl>,  
## # seed\_type <dbl>, region <dbl>, dummy <dbl>, afff\_factor <dbl>,  
## # degg\_factor <dbl>, comp\_count\_list <list>, comp\_count\_pre <dbl>,  
## # comp\_count\_current <dbl>, mirna\_count\_list <list>,  
## # mirna\_count\_pre <dbl>, mirna\_count\_current <dbl>,  
## # mirna\_count\_per\_dep <dbl>, effect\_current <dbl>, effect\_pre <dbl>,  
## # effect\_list <list>, mirna\_count\_per\_comp <dbl>

When the graphs which were resulted from analyses were examined, it was observed that behaviours were same. But, when the results were analysed in terms of expression values, the regulation differences can be observed.

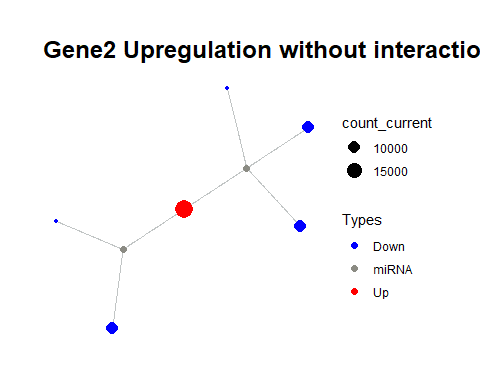
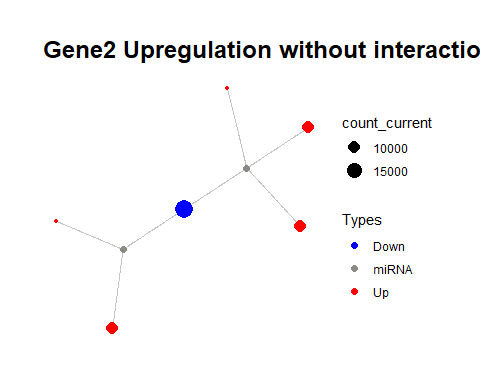
priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)%>%  
 update\_how("Gene2", 2)%>%  
 simulate(3)%>%  
 activate(edges)%>%  
 as\_tibble()%>%  
 select(Competing\_name,comp\_count\_list,effect\_list)%>%  
 unnest()

## # A tibble: 28 x 3  
## Competing\_name comp\_count\_list effect\_list  
## <chr> <dbl> <dbl>  
## 1 Gene1 10000 263.   
## 2 Gene1 10065. 198.   
## 3 Gene1 10064. 199.   
## 4 Gene1 10064. 199.   
## 5 Gene2 10000 6.58  
## 6 Gene2 19997. 9.91  
## 7 Gene2 19997. 9.88  
## 8 Gene2 19997. 9.88  
## 9 Gene3 5000 91.5   
## 10 Gene3 5023. 68.8   
## # ... with 18 more rows

## Common target perturbation in *minsamp* dataset.

There are hundreds of defined miRNAs for human, so this results in presence of common targets of miRNAs in cells. Therefore, we have analysed perturbation efficiency of common target in *minsamp* dataset.

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)%>%  
 update\_how("Gene4", 2)%>%  
 simulate\_vis(Competing\_color = "navajowhite3", mirna\_color = "ivory4", Upregulation = "red", title = "Gene2 Upregulation without interaction factors",2)



## # A tbl\_graph: 8 nodes and 7 edges  
## #  
## # A rooted tree  
## #  
## # Node Data: 8 x 7 (active)  
## name type node\_id initial\_count count\_pre count\_current  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Gene1 Comp~ 1 10000 10028. 10027.  
## 2 Gene2 Comp~ 2 10000 10001. 10001.  
## 3 Gene3 Comp~ 3 5000 5010. 5009.  
## 4 Gene4 Comp~ 4 10000 19803. 19806.  
## 5 Gene5 Comp~ 5 5000 5024. 5024.  
## 6 Gene6 Comp~ 6 10000 10044. 10044.  
## # ... with 2 more rows, and 1 more variable: changes\_variable <chr>  
## #  
## # Edge Data: 7 x 23  
## from to Competing\_name miRNA\_name Competing\_expre~ miRNA\_expression  
## <int> <int> <chr> <chr> <dbl> <dbl>  
## 1 1 7 Gene1 Mir1 10000 1000  
## 2 2 7 Gene2 Mir1 10000 1000  
## 3 3 7 Gene3 Mir1 5000 1000  
## # ... with 4 more rows, and 17 more variables: energy <dbl>,  
## # seed\_type <dbl>, region <dbl>, dummy <dbl>, afff\_factor <dbl>,  
## # degg\_factor <dbl>, comp\_count\_list <list>, comp\_count\_pre <dbl>,  
## # comp\_count\_current <dbl>, mirna\_count\_list <list>,  
## # mirna\_count\_pre <dbl>, mirna\_count\_current <dbl>,  
## # mirna\_count\_per\_dep <dbl>, effect\_current <dbl>, effect\_pre <dbl>,  
## # effect\_list <list>, mirna\_count\_per\_comp <dbl>

The common target perturbation (increasing to two fold at Gene4 expression in presence of interaction factors) resulted in more prominent efficiency at the same conditions (shown in following).

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)%>%  
 update\_how("Gene4", 2)%>%  
 simulate(3)%>%  
 activate(edges)%>%  
 as\_tibble()%>%  
 select(Competing\_name,comp\_count\_list,effect\_list)%>%  
 unnest()

## # A tibble: 28 x 3  
## Competing\_name comp\_count\_list effect\_list  
## <chr> <dbl> <dbl>  
## 1 Gene1 10000 263.   
## 2 Gene1 10028. 236.   
## 3 Gene1 10027. 237.   
## 4 Gene1 10027. 237.   
## 5 Gene2 10000 6.58  
## 6 Gene2 10001. 5.89  
## 7 Gene2 10001. 5.90  
## 8 Gene2 10001. 5.90  
## 9 Gene3 5000 91.5   
## 10 Gene3 5010. 81.9   
## # ... with 18 more rows

## Determination of perturbation efficiencies efficiencies of elements in system.

priming\_graph(minsamp, competing\_count = Competing\_expression, miRNA\_count = miRNA\_expression, aff\_factor = c(energy, seed\_type), deg\_factor = region)-> sample\_graph  
  
find\_node\_perturbation(sample\_graph, how = 2, cycle = 3, limit = 0.1)

## # A tibble: 8 x 9  
## name type node\_id initial\_count count\_pre count\_current  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Gene1 Comp~ 1 10000 10000 10000  
## 2 Gene2 Comp~ 2 10000 10000 10000  
## 3 Gene3 Comp~ 3 5000 5000 5000  
## 4 Gene4 Comp~ 4 10000 10000 10000  
## 5 Gene5 Comp~ 5 5000 5000 5000  
## 6 Gene6 Comp~ 6 10000 10000 10000  
## 7 Mir1 miRNA 7 1000 1000 1000  
## 8 Mir2 miRNA 8 2000 2000 2000  
## # ... with 3 more variables: changes\_variable <chr>,  
## # perturbation\_efficiency <dbl>, perturbed\_count <dbl>

# 2. Obtaining breast cancer dataset and integration

This section describes how to apply ceRNAnetsim package on a breast cancer patient miRNA:target interaction dataset. Before the approach, we obtained three datasets and combined them.

## 2.1 How to get gene expression counts of TCGA-E9-A1N5 patient.

We have obtained the gene expression values of patient using [TCGAbiolinks](https://www.bioconductor.org/packages/release/bioc/html/TCGAbiolinks.html) package from [Bioconductor](https://www.bioconductor.org/). For this process, we have followed the instructions of the package. TCGAbiolinks package provides to obtain data for whole number of given barcode(s) at once. But, we preferred to download them seperately to show datasets.

* Obtain to gene expression counts of tumor tissue.

BCP\_tumor <- GDCquery(project = "TCGA-BRCA",  
 data.category = "Transcriptome Profiling",  
 data.type = "Gene Expression Quantification",   
 workflow.type = "HTSeq - Counts",  
 barcode = "TCGA-E9-A1N5-01A-11R-A14D-07")  
  
GDCdownload(BCP\_tumor)  
BCPGE\_tumor <- GDCprepare(BCP\_tumor)  
  
as.data.frame(assay(BCPGE\_tumor))%>%  
 mutate(ensembl\_gene\_id = rownames(.))%>%  
 dplyr::inner\_join(as.data.frame(rowData(BCPGE\_tumor)), by = "ensembl\_gene\_id")%>%  
 dplyr::select(ensembl\_gene\_id, external\_gene\_name, 1)-> TCGA\_E9\_A1N5\_tumor  
  
colnames(TCGA\_E9\_A1N5\_tumor)[3] <- "GE\_tumor"  
  
head(TCGA\_E9\_A1N5\_tumor)

* Obtain to gene expression counts of normal tissue.

BCP\_normal <- GDCquery(project = "TCGA-BRCA",  
 data.category = "Transcriptome Profiling",  
 data.type = "Gene Expression Quantification",   
 workflow.type = "HTSeq - Counts",  
 barcode = "TCGA-E9-A1N5-11A-41R-A14D-07")  
GDCdownload(BCP\_normal)  
  
BCPGE\_normal <- GDCprepare(BCP\_normal)  
  
as.data.frame(assay(BCPGE\_normal))%>%  
 mutate(ensembl\_gene\_id = rownames(.))%>%  
 dplyr::inner\_join(as.data.frame(rowData(BCPGE\_normal)), by = "ensembl\_gene\_id")%>%  
 dplyr::select(ensembl\_gene\_id, external\_gene\_name, 1)-> TCGA\_E9\_A1N5\_normal  
  
colnames(TCGA\_E9\_A1N5\_normal)[3] <- "GE\_normal"  
  
head(TCGA\_E9\_A1N5\_normal)

## 2.2 How to get miRNA expression counts of TCGA-E9-A1N5 patient.

We have used TCGAbiolinks package to obtain miRNA expression quantification. The query gives read count of miRNA as isoform chromosome coordination. The data also contains mature miRNA information. So, we processed data to attain -5p -3p isoform information using mirbase release21 dataset.

* Get the mirbase id of mature miRNA:

We downloaded the mirbase release 21 dataset from [mirbase](http://www.mirbase.org/ftp.shtml) and processed the patient mirna expression datasets as following:

library(readr)  
read\_tsv("hsa\_mirna.txt", comment = "#", col\_names = FALSE)%>%  
 dplyr::select(mirna\_type= X3, definition = X9)%>%  
 filter(!endsWith(mirna\_type, "primary\_transcript"))%>%  
 tidyr::separate(definition, c("ID", "Alias", "Name", "Derivated"), sep = ";")%>%  
 dplyr::select(Alias, Name)%>%  
 tidyr::separate(Alias, c("trash1", "ID"), sep = "=")%>%  
 tidyr::separate(Name, c("trash2", "Name"), sep = "=")%>%  
 dplyr::select(-trash1, -trash2)-> mirbase\_id\_conv  
  
head(mirbase\_id\_conv)

## # A tibble: 6 x 2  
## ID Name   
## <chr> <chr>   
## 1 MIMAT0027618 hsa-miR-6859-5p  
## 2 MIMAT0027619 hsa-miR-6859-3p  
## 3 MIMAT0005890 hsa-miR-1302   
## 4 MIMAT0027618 hsa-miR-6859-5p  
## 5 MIMAT0027619 hsa-miR-6859-3p  
## 6 MIMAT0049032 hsa-miR-12136

* Obtain the miRNA expression of tumor tissue of patient:

BCP\_mirnatumor <- GDCquery(project = "TCGA-BRCA",  
 data.category = "Transcriptome Profiling",  
 data.type = "Isoform Expression Quantification",  
 workflow.type = "BCGSC miRNA Profiling",  
 barcode = "TCGA-E9-A1N5-01A-11R-A14C-13")  
  
GDCdownload(BCP\_mirnatumor)  
  
GDCprepare(BCP\_mirnatumor)%>%  
 as.data.frame()%>%  
 dplyr::select(miRNA\_ID, read\_count, reads\_per\_million\_miRNA\_mapped, miRNA\_region)%>%  
 dplyr::filter(startsWith(miRNA\_region, "mature"))%>%  
 dplyr::mutate(mirbase\_id =str\_remove(miRNA\_region, "mature,"))%>%  
 dplyr::select(-miRNA\_region)%>%  
 dplyr::inner\_join(mirbase\_id\_conv, by = c("mirbase\_id"="ID"))%>%  
 dplyr::select(miRNA\_name = Name, read\_count, reads\_per\_million\_miRNA\_mapped)%>%  
 dplyr::group\_by(miRNA\_name)%>%  
 mutate(read\_count= sum(read\_count), reads\_per\_million\_miRNA\_mapped = sum(reads\_per\_million\_miRNA\_mapped))%>%  
 dplyr::ungroup()%>%  
 distinct() -> BCPME\_mirnatumor  
  
head(BCPME\_mirnatumor)

* Obtain the miRNA expression of normal tissue of patient:

BCP\_mirnanormal <- GDCquery(project = "TCGA-BRCA",  
 data.category = "Transcriptome Profiling",  
 data.type = "Isoform Expression Quantification",  
 workflow.type = "BCGSC miRNA Profiling",  
 barcode = "TCGA-E9-A1N5-11A-41R-A14C-13")  
  
GDCdownload(BCP\_mirnanormal)  
# a616435d-0b69-48ac-813d-5d75ad9b85eb.mirbase21.isoforms.quantification.txt  
  
GDCprepare(BCP\_mirnanormal)%>%  
 as.data.frame()%>%  
 dplyr::select(miRNA\_ID, read\_count, reads\_per\_million\_miRNA\_mapped, miRNA\_region)%>%  
 dplyr::filter(startsWith(miRNA\_region, "mature"))%>%  
 dplyr::mutate(mirbase\_id =str\_remove(miRNA\_region, "mature,"))%>%  
 dplyr::select(-miRNA\_region)%>%  
 dplyr::inner\_join(mirbase\_id\_conv, by = c("mirbase\_id"="ID"))%>%  
 dplyr::select(miRNA\_name = Name, read\_count, reads\_per\_million\_miRNA\_mapped)%>%  
 dplyr::group\_by(miRNA\_name)%>%  
 mutate(read\_count= sum(read\_count), reads\_per\_million\_miRNA\_mapped = sum(reads\_per\_million\_miRNA\_mapped))%>%  
 dplyr::ungroup()%>%  
 distinct() -> BCPME\_mirnanormal  
  
head(BCPME\_mirnanormal)

## 2.3 Get the high-throughput experimental miRNA:target dataset.

There are various datasets about miRNA:target pairs such as miRTarBase, DianaTools, miRecords, miRWalk etc. Some of these present the experimentally supported miRNA target pairs or only predicted ones. The experimentally supported datasets generally provides weak evidence for interactions. For these reasons, we obtained the high-throughput experimental miRNA:target dataset from two studies performed by Helwak et al. and Moore et al. These steps were not handle in this file because they contain many processes.

Briefly these datasets contain various common information about miRNA:target interactions such as the miRNA name, miRNAsequence, target name, target sequence, their chromosomal locations, binding location on the target sequence, binding free energy, seed structure. But these datasets provides the informations with different data structures.So we followed the steps:

* The datasets were directly downloaded from supplementary data files of the studies.
* It was provided that the datasets are converted to same human genome build.
* The seed type information was organized as the same style.
* The datasets were combined.
* We commited the interaction factors as numeric values according to previous studies. (We added the interaction factors and their numeric values at Supplementary tables S2-3)

Finally, we have obtained the experimentally supported miRNA:target dataset.

data("experimentalmirnagene")  
  
head(experimentalmirnagene)

## # A tibble: 6 x 18  
## cluster chromosome start\_position end\_position strand hgnc\_symbol  
## <chr> <chr> <int> <int> <chr> <chr>   
## 1 0727A-~ chr5 162864575 162873157 1 CCNG1   
## 2 L1HS-1~ chr14 95552565 95624347 -1 DICER1   
## 3 L2HS-8~ chr6 109307640 109416022 -1 SESN1   
## 4 L2HS-1~ chr5 36876861 37066515 1 NIPBL   
## 5 L2-407~ chr4 106603784 106817143 -1 INTS12   
## 6 L1HS-7~ chr5 130977407 131132710 -1 FNIP1   
## # ... with 12 more variables: Ensembl\_Gene\_Id <chr>,  
## # Ensembl\_Transcript\_Id <chr>, target\_seq <chr>, miRNA <chr>,  
## # miR\_seq <chr>, seed\_type <chr>, Energy <dbl>, HG38build\_loc <chr>,  
## # Genom\_build <chr>, region <chr>, region\_effect <dbl>,  
## # seed\_type\_effect <dbl>

The methods about miRNA:target interactions are based a basic principle that is reading after isolation of miRNA:target chimerics. The datasets contain all the chimeric miRNA:target structures found in the medium during the experiment. On the other hand, it could be said that the reading is performed as snapshot.Because of that, the methods can provide different chimeric interactions the same miRNA:target pair. We have preferred to select most effective interaction parameters for the same miRNA:target pairs that can exhibit various interactions. The step is performed as:

experimentalmirnagene%>%  
 dplyr::select(miRNA, Ensembl\_Gene\_Id, hgnc\_symbol, Energy, seed\_type\_effect, region\_effect)%>%  
 distinct()%>%  
 group\_by(Ensembl\_Gene\_Id, miRNA)%>%  
 mutate(seed\_type\_effect= ifelse(seed\_type\_effect==max(seed\_type\_effect), seed\_type\_effect, max(seed\_type\_effect)), Energy = ifelse(Energy==min(Energy), Energy, min(Energy)), region\_effect= ifelse(region\_effect==max(region\_effect), region\_effect, max(region\_effect)))%>%  
 distinct()-> tocombine\_mirnagene  
  
head(tocombine\_mirnagene)

## 2.4 Combine the dataset

BCPME\_mirnanormal%>%  
 dplyr::inner\_join(tocombine\_mirnagene, by = c("miRNA\_name"="miRNA"))%>%  
 dplyr::inner\_join(TCGA\_E9\_A1N5\_normal, by = c("Ensembl\_Gene\_Id"="ensembl\_gene\_id", "hgnc\_symbol"="external\_gene\_name"))%>%  
 distinct()%>%  
 dplyr::select(hgnc\_symbol, miRNA\_name, mirna\_RPM= reads\_per\_million\_miRNA\_mapped, GE\_normal, Energy, seed\_type\_effect, region\_effect)-> E9GE\_mirnagenenormal

data("E9GE\_mirnagenenormal")  
head(E9GE\_mirnagenenormal)

## # A tibble: 6 x 7  
## hgnc\_symbol miRNA\_name mirna\_RPM GE\_normal Energy seed\_type\_effect  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 CCNG1 hsa-let-7~ 111204. 5245 -25.1 0.05  
## 2 DICER1 hsa-let-7~ 111204. 3285 -24.4 0.43  
## 3 SESN1 hsa-let-7~ 111204. 1179 -22.2 0.05  
## 4 NIPBL hsa-let-7~ 111204. 4503 -22.1 0.05  
## 5 INTS12 hsa-let-7~ 111204. 600 -21.9 0.05  
## 6 FNIP1 hsa-let-7~ 111204. 1248 -21.8 0.43  
## # ... with 1 more variable: region\_effect <dbl>

BCPME\_mirnatumor%>%  
 dplyr::inner\_join(tocombine\_mirnagene, by = c("miRNA\_name"="miRNA"))%>%  
 dplyr::inner\_join(TCGA\_E9\_A1N5\_tumor, by = c("Ensembl\_Gene\_Id"="ensembl\_gene\_id", "hgnc\_symbol"="external\_gene\_name"))%>%  
 distinct()%>%  
 dplyr::select(hgnc\_symbol, miRNA\_name, mirna\_RPM= reads\_per\_million\_miRNA\_mapped, GE\_tumor, Energy, seed\_type\_effect, region\_effect)-> E9GE\_mirnagenetumor

data("E9GE\_mirnagenetumor")  
head(E9GE\_mirnagenetumor)

## # A tibble: 6 x 7  
## hgnc\_symbol miRNA\_name mirna\_RPM GE\_tumor Energy seed\_type\_effect  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 CCNG1 hsa-let-7~ 62406. 2467 -25.1 0.05  
## 2 DICER1 hsa-let-7~ 62406. 5023 -24.4 0.43  
## 3 SESN1 hsa-let-7~ 62406. 829 -22.2 0.05  
## 4 NIPBL hsa-let-7~ 62406. 5126 -22.1 0.05  
## 5 INTS12 hsa-let-7~ 62406. 1009 -21.9 0.05  
## 6 FNIP1 hsa-let-7~ 62406. 2144 -21.8 0.43  
## # ... with 1 more variable: region\_effect <dbl>

## 2.5 Selection of trigger node

We have compared two datasets that are obtained for the tumor and normal tissue samples of same patient. We tried to change expression of a gene in normal tissue as the same level in the tumor tissue.

For this step, we have determined the changes of the gene expression in terms of fold change:

E9GE\_mirnagenetumor%>%  
 dplyr::select(hgnc\_symbol, GE\_tumor)%>%  
 dplyr::inner\_join((E9GE\_mirnagenenormal%>%dplyr::select(hgnc\_symbol, GE\_normal)), by ="hgnc\_symbol")%>%  
 dplyr::mutate(FC= GE\_tumor/GE\_normal)%>%  
 distinct()%>%  
 filter(FC>2.5, FC<3.5)-> three\_fold\_change  
  
# ABCC1 gene has 5827 read count in tumor tissue although 1420 in normal tissue (FC=4.10)

Secondly, we have determined the most important nodes of network. We applied find\_node\_perturbation function. We only defined nodes that affect the other nodes more than 1.05 fold change with 10 iteration when they increase 3 fold.

This dataset, perturbationofnetwork, includes 382 genes of totally 423 nodes.

Result of this, we obtained common nodes of these two datasets (i.e perturbationofnetwork and three\_fold\_change) and selected a gene, SERPINE2.

three\_fold\_change%>%  
 inner\_join(perturbationofnetwork, by = c("hgnc\_symbol"="name"))  
  
#Selected node is SERPINE2

## 2.5 Approach of Method into Combined Datasets

We selected SERPINE2 gene for simulation of regulation on network.

### 2.5.1 Find iteration of simulation

as.data.frame(E9GE\_mirnagenenormal)%>%  
 priming\_graph(competing\_count = GE\_normal, miRNA\_count = mirna\_RPM, aff\_factor = c(Energy, seed\_type\_effect), deg\_factor = region\_effect)%>%  
 update\_how("SERPINE2",2.75) %>%  
 simulate(150) %>%  
 find\_iteration(limit=1, plot= TRUE) #limit=1 describes the change that is not taken into account.

The node amount of changed gene on the system in terms of percentage were shown in above. As seen, firstly, the changed gene count increase. The system which contains the hundreds of miRNAs and thousands of genes can slowly gain the steady-state again. At first glance, it can be assumed that when all nodes in the system are reached, stable state will be provided. However, although all nodes are reached, the nodes competing with each other cause the edits to continue for a while.

The dynamics of the approach are shown in package viggnettes [link](https://selcenari.github.io/ceRNAnetsim/articles/convenient_iteration.html).

So, we offered an approach about to find iteration. find\_iteration function does not give the iteration to gain steady-state, but it gives the iteration which has maximum affected node counts. The function is applied as following:

as.data.frame(E9GE\_mirnagenenormal)%>%  
 priming\_graph(competing\_count = GE\_normal, miRNA\_count = mirna\_RPM, aff\_factor = c(Energy, seed\_type\_effect), deg\_factor = region\_effect)%>%  
 update\_how("SERPINE2",2.75) %>%  
 simulate(100) %>%  
 find\_iteration(limit=1, plot= FALSE)  
  
#31

### 2.5.2 Simulation of dataset

We tried to apply two fold of the point that SERPINE2 has maximum affected genes on network.

as.data.frame(E9GE\_mirnagenenormal)%>%  
 priming\_graph(competing\_count = GE\_normal, miRNA\_count = mirna\_RPM, aff\_factor = c(Energy, seed\_type\_effect), deg\_factor = region\_effect)%>%  
 update\_how("SERPINE2",2.75)%>%  
 simulate(62)

as.data.frame(E9GE\_mirnagenenormal)%>%  
 priming\_graph(competing\_count = GE\_normal, miRNA\_count = mirna\_RPM, aff\_factor = c(Energy, seed\_type\_effect), deg\_factor = region\_effect)%>%  
 update\_how("SERPINE2",2.75)%>%  
 simulate(62)%>%  
 as\_tibble()%>%  
 select(name, initial\_count, count\_current)->simulation\_results

### 2.5.3 Comparison of simulation results and tumor tissue expression values

E9GE\_mirnagenetumor%>%  
 dplyr::select(hgnc\_symbol, GE\_tumor)%>%  
 dplyr::inner\_join((E9GE\_mirnagenenormal%>%dplyr::select(hgnc\_symbol, GE\_normal)), by ="hgnc\_symbol")%>%  
 inner\_join(simulation\_results, by= c("hgnc\_symbol"="name"))%>%  
 distinct()

Actually, we have developed to provide a new approach mirna mediated regulation networks. This approach may not explain the whole regulation behaviors between miRNAs and targets but can be first step to more detailed and cohorent miRNA:target regulation approach.

# 3. REFERENCES