# *f*RNC Tutorial

# Methods

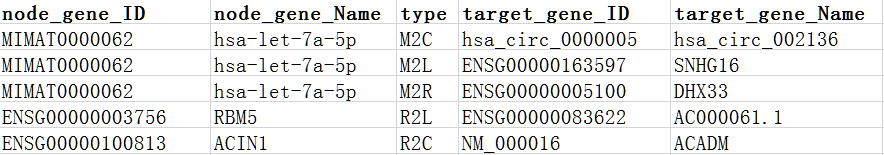
**Construction of the basal RBP-ncRNAs network**

The mutual interactions among RBPs and ncRNAs (miRNAs, circRNAs and lncRNAs) were extracted from ENCORI database across 2 species (human, mouse) (http://starbase.sysu.edu.cn/). Each interaction has a stringency value (>=1, low; >=2 medium; >=3 high; >=5 strict), where the values indicated the supported CLIP/PARE experimental evidence. And the RBP-RBP interactions were extracted from STRING database[[19](#_ENREF_19)]. Each interaction has a confidence score (>=0.15, low; >=0.4 medium; >=0.7 high; >=0.9 strict) as described in the FAQ of the database STRING(https://string-db.org/). Meanwhile, RBP-miRNA interactions were downloaded from the RNAct database (https://rnact.crg.eu/)[[20](#_ENREF_20)]. The null distribution of RBP-miRNA interaction score are with mean (Human: 4.88841; Mouse: 4.346568) and standard deviation (Human:5.72845; Mouse:4.636672). So we set the threshold of RBP-miRNA interaction with stringency value (>=0, low; >=5 medium; >=10 high; >=20 strict). There are 7 types of non-directional interactions in the regulatory network: miRNA-lncRNA (M2L), miRNA-circRNA (M2C), miRNA-RBP (M2R), RBP-lncRNA (R2L), RBP-miRNA (R2M), RBP-RBP (R2R) and RBP-circRNA (R2C). As a result, 1934(1431) miRNAs, 14281(9129) lncRNAs, 91280(9660) circRNAs and 1297(1876) RBPs constitute 11788434 interactions in the human(mouse). Table 1 lists the interactions statistics the RBP-ncRNA network.

Table 1: Summeary of the RBP-ncRNA regulation network in *f*RNC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| human(mouse) | Interaction | miRNA | circRNA | lncRNA | RBP |
| miRNA-circRNA | 3502309(87483) | 642(694) | 83324(6342) |  |  |
| miRNA-lncRNA | 63698(10468) | 642(696) |  | 3789(826) |  |
| miRNA-mRNA(RBP) | 125409(126534) | 618(676) |  |  | 1271(1743) |
| RBP-circRNA | 5217528(533553) |  | 91280(9660) |  | 132(37) |
| RBP-lncRNA | 201179(34881) |  |  | 14281(9129) | 132(40) |
| RBP-RBP | 169913(255905) |  |  |  | 1267(1876) |
| RBP-miRNA | 2508398(2585817) | 1934(1431) |  |  | 1297(1807) |
| All | 11788434(3634641) | 1934(1431) | 91280(9660) | 14281(9129) | 1297(1876) |

Users can also provide their own interaction data. The following input format is accepted.

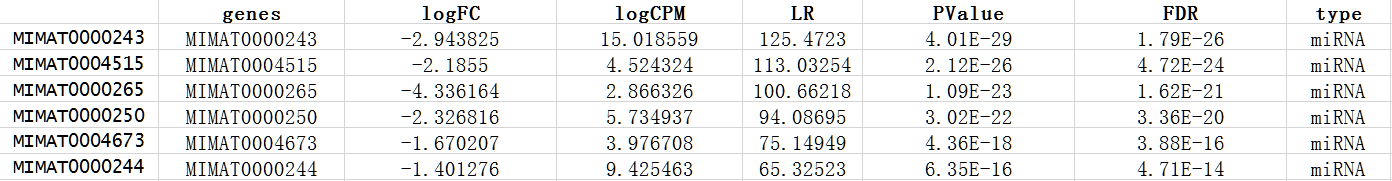


During the construction of regulatory network, the lncRNA and gene names were mapped to Ensembl IDs (http://asia.ensembl.org/index.html), the miRNA names were mapped to miRBase accession numbers of mature miRNAs (http://www.mirbase.org/) and the circRNA names were mapped to circBase ID (http://www.circbase.org/). The RBPs, miRNAs, lncRNAs, and circRNA that did not map to these IDs were discarded.

**Expression data pre-processing**

The R package *fRNC* can process the expression data of ncRNAs and mRNA in matched tumor and normal samples from either microarray or RNA-seq platform. For instance, we first download cancer miRNA, mRNA and protein expression data from the TCGA database and TPCA database(https://portal.gdc.cancer.gov/, https://www.tcpaportal.org/tcpa/download.html). Then the *DEGs* function can use the R package *edgeR* to perform differential expression analysis for RNA-seq count data. For miRNA expression, we calculated the read counts of each mature miRNA from the isoform quantification files, in which mature miRNA IDs are updated based on miRBase release 22 (http://www.mirbase.org/). Genes in these samples were [annotate](javascript:;)d by the Ensembl 90 annotation of the human genome. The lncRNA IDs were [extract](javascript:;)ed based on the gene biotype in the annotation information. The circRNA IDs are updated by extracting the overlap of circRNA transcripts that are from both MiOncoCirc and circBase. Furthermore, the miRNAs, lncRNAs, RBP (mRNA), and circRNAs that expressed (expression value >0) in at least half of the samples were retained. Some examples of ncRNA and mRNA names can be seen in the following examples.

Users can also provide data after differential expression. The input format is as follows. The column name must contain genes, log fold change (logFC, optional), *P* Value and type. The row name is the gene identifier.



**Survival analysis**

The corresponding clinical data such as survival time can also be used. The *fRNC* package evaluates each gene (node) by calculating the *P*-values from a univariable Cox proportional hazards regression model with the R package *survival*, which quantifies how significantly the expression data of the node correlated with the corresponding patient clinical data.

**Identification of the dysregulated RNCs**

The fRNC package contains two sub-network searching algorithms: fast exact and greedy search approach. When users choose the method of "global", the fast exact approach was used to calculate high-scoring RNC from the whole network. The following depicts a short outline of the algorithm:

1. Given a set of P values from either differentially analysis or survival analysis, they are first transformed to a node score as proposed in the R package BioNet. Then all positive connected nodes are aggregated into meta-nodes based on the different connected sub-networks decomposed from the whole network.

2. The node scores are transformed to the edge scores based on the node that an edge linked.

3. All paths between positive meta-nodes are calculated based on the minimum spanning tree (MST) of these edge scores. It can include negative scoring nodes.

4. The highest sub-networks score is calculated on an MST, in which the negative nodes and positive meta-nodes were connected.

For global method, the summary score for a RNC is defined in as

 (1)

where *k* denotes the number of nodes in the network. ∑*z*iindicates the sum of these edge scores in the network*.*

When users choose the method of "local", the searching algorithm is a greedy search approach. Briefly, each node scorewas calculated from *P* value according to *z*node = Φ-1(1 - *p*i) (*p* indicated the significance of expression change), The *φ*-1 represents the inverse normal cumulative distribution function. Secondly, the edge score was calculated according to the change of gene co-expression in case and control samples using the following equations.

 (2)

 (3)

 (4)

Subsequently, the sub-network score is the sum of node scores and edge scores as follows:

 (5)

where *nnode* and *nedge* denote the number of nodes and edges in the sub-network. The parameter α∈(0,1) is used to control the weight of node score and edge score(here α=0.5).

We implemented a greedy algorithm as follows:

1. Assigning a seed node by the user. Firstly, the sub-network only contains the seed node. Zall of the seed sub-network is computed.

2. Identifying neighborhood nodes, whose shortest path to any node in the sub-network is shorter or equal to a distance *d* (e.g., *d* = 2). In other words, identifying the all connected neighborhood nodes with a distance *d* around the sub-network.

3. Exploring the neighborhood nodes defined in 2 and finding which nodes can generate the maximum increment of Zm. If the increment (Zm+1 - Zm) is greater than Zm×*r,* the nodes will be added into the sub-network, where *r* is the rate of proportion increment (Zm+1 > Zm×(1+*r*)). Here, the *r* is set to 0.1.

4. Keeping going 2‐3 until any neighborhood nodes added into the current sub-network cannot yield an increment that is greater than Zm×*r*.

Lastly, to evaluate the statistical significance of the sub-network identified by the two searching algorithms. It uses random sampling to see if the score of the sub-network is significantly higher than that of a random node set in the network. In this way, we randomly sample node sets with the same number 100 000 times. The mean μn and standard deviation σn are estimated from the sampled nodes. Finally, the sub-network (*i.e.* RNC) score is evaluated further normalized by



The *fRNC* package contains two sub-network searching algorithms: fast exact and greedy search approach. For both methods, the summary score for a RNC is defined in as



where *k* denotes the number of nodes in the network. *z*iindicates the score of the node*.*

When users choose the method of "global", the fast exact approach was used to calculate high-scoring RNC from the whole network. The following depicts a short outline of the algorithm:

1. Given a set of P values from either differentially analysis or survival analysis, they are first transformed to a node score as proposed in R package *BioNet*. Then all positive connected nodes are aggregated into meta-nodes based on the different connected sub-networks decomposed from the whole network.

2. The node scores are transformed to the edge scores based on the node that an edge linked.

3. All paths between positive meta-nodes are calculated based on the minimum spanning tree (MST) of these edge scores. It can include negative scoring nodes.

4. The highest sub-networks score is calculated on a MST, in which the negative nodes and positive meta-nodes were connected.

When users choose the method of "local", the searching algorithm is greedy search approach. Briefly, the score of *z*iwas calculated from *P* value according to *z*i = Φ-1(1 - *p*i) (*p* indicated the significance of expression change determined by*edgeR* package), The *φ*-1 represents the inverse normal cumulative distribution function.

1. Assigning a seed node by the user. Firstly, the sub-network only contains the seed node. Zm of the seed sub-network is computed.

2. Identifying neighborhood nodes, whose shortest path to any node in the sub-network is shorter or equal to a distance *d* (e.g., *d* = 2). In other words, identifying the all connected neighborhood nodes with a distance *d* around the sub-network.

3. Exploring the neighborhood nodes defined in 2 and find which nodes can generate the maximum increment of Zm. If the increment (Zm+1 - Zm) is greater than Zm×*r,* the nodes will be added into the sub-network, where *r* is the rate of proportion increment (Zm+1 > Zm×(1+*r*)). Here, the *r* is set to 0.1.

4. Keeping going 2‐3 until any neighborhood nodes added into the current sub-network cannot yield an increment that is greater than Zm×*r*.

Lastly, to evaluate the statistical significance of the sub-network identified by the two searching algorithm. It uses random sampling to see if the score of the sub-network is significantly higher than that of a random node set in the network. In this way, we randomly samples node sets with the same number 100 000 times. The mean μn and standard deviation σn are estimated from the sampled nodes. Finally, sub-network (*i.e.* RNC) score is evaluated further normalized by



# Example

**Esophageal carcinoma data collection and pre-processing**

We downloaded miRNA, mRNA expression data of esophageal carcinoma (ESCA) and normal samples from TCGA. The miRNA, mRNA-seq data consist of 161 tumors with 11 normal samples in the TCGA. The miRNA-lncRNA and miRNA-mRNA(RBP) interactions were extracted from ENCORI with a high stringency, where the number of the supported CLIP experimental evidence is 3 or greater. The R package *edgeR* was used to analyze the differentially expressed miRNAs, mRNAs and lncRNAs.

**Case 1**

The major aim of this analysis is to detect ncRNA RNC by the "global" method from miRNA-lncRNA-RBP regulation network in ESCA.

First of all, we loaded the miRNA, mRNA expression data of the ESCA cancer and normal samples separately. Then we loaded the survival data. *DEGs* function was used to identify the differentially expressed miRNAs and mRNAs, and to assign the type attribute to each node. By setting the parameter 'data\_type', the *DEGs* function can handle expression data generated from either sequencing or microarray platform.

> library(fRNC)

> data("case.exp\_miRNA")

> data("control.exp\_miRNA")

> result\_miR <- DEGs(case.exp\_miRNA,control.exp\_miRNA, geneid= rownames(control.exp\_miRNA), data\_type = "RNAseq\_counts")

> data("case.exp\_rna")

> data("control.exp\_rna")

> result\_rna <- DEGs(case.exp\_rna,control.exp\_rna, geneid= rownames(control.exp\_rna), data\_type = "RNAseq\_counts")

Then, we extract interactions according to stringency with number of supported experiments (>=1, low; >=2 medium; >=3 high; >=5 strict) and interaction type with either the RBP-ncRNA network, which belonged to the category " transcriptome". Here, we set stringency with "high" and type "transcription " in the function *interStringency*.

>interac\_cerna <- interStringency(type = "transcription", spec ="hg",stringency = "high")

> interac\_cerna <- interac\_cerna[,c("node\_gene\_ID","type","target\_gene\_ID")]

Meanwhile, we need to generate corrected p-value based on p-value and logFC with the formula: *2 × (1- log10P-value × |logFC|)* (the parameter *islog = TRUE* ), and deal with unreasonable p-value in the function *combinp*. Thus nodes with zero p-values are assigned to the minimum p-value of the remaining nodes. The nodes with p-value of 1 are deleted directly.

> nt <- dim(case.exp\_miRNA)[2]

> nn <- dim(control.exp\_miRNA)[2]

> case\_exp <- rbind(result\_miR$Nor\_expr[,1:nt],result\_rna$Nor\_expr[,1:nt])

> control\_exp <- rbind(result\_miR$Nor\_expr[,(nt+1):(nt+nn)],result\_rna$Nor\_expr[,(nt+1):(nt+nn)])

> dataNo <- rbind(result\_miR$DEGs, result\_rna$DEGs)

> head(dataNo)

genes logFC logCPM LR PValue FDR type

MIMAT0000243 MIMAT0000243 -2.943825 15.018559 125.47230 4.011512e-29 1.785123e-26 miRNA

MIMAT0004515 MIMAT0004515 -2.185500 4.524324 113.03254 2.122628e-26 4.722847e-24 miRNA

MIMAT0000265 MIMAT0000265 -4.336164 2.866326 100.66218 1.090888e-23 1.618150e-21 miRNA

MIMAT0000250 MIMAT0000250 -2.326816 5.734937 94.08695 3.019632e-22 3.359340e-20 miRNA

MIMAT0004673 MIMAT0004673 -1.670207 3.976708 75.14949 4.363893e-18 3.883865e-16 miRNA

MIMAT0000244 MIMAT0000244 -1.401276 9.425463 65.32523 6.350310e-16 4.709813e-14 miRNA

> gene2weight <- combinp(dataNo[,c("type","logFC","PValue")], islog = T)

Now we used a fast exact approach to calculate the maximum scoring sub-network to identify the "global" RNC with the parameter of "global" in the main function by a FDR of 1e-5. The smaller the FDR was set, the smaller the sub-network can be obtained. Such a low FDR was chosen to obtain a small sub-network.

> res.list\_global<- runmodule(network = interac\_cerna, gene2weight, method = "global", FDR = 1e-5)

>names(res.list\_global)

"GNCW" "module" "module.score.matrix"

#GNCW, an object of igraph class, is the node‐weighted ncRNA network.

#module, an object of igraph class, contains the gene in GNCW and the weight

#module.score.matrix, an object of matrix, contains data for the module: Zm (module score),

Zn (normalized module score). The higher the zi value is, the more reliable of the module it is.

We saved the result as XGMML file and then observed it in the Cytoscape environment, where coloring and changing the node shape can be manipulated according to the "type".

>saveNetwork(res.list\_global$module,file="ceRNA\_module",type = "XGMML")

Following figure is the dysregulated RNC in ESCA ncRNA network. Red [circle](javascript:;), blue diamond, and grey rectangle nodes represent RBPs, lncRNAs, and miRNAs, respectively.

**未标题-1.tif**

**Figure 1. The "global" RNC identified for ncRNAs network in ESCA**

**Case 2**

The major aim of this analysis is to detect RNC containing one RNA binding protein “SRSF1” by the "local" method from miRNA-lncRNA-RBP regulation network in BRCA.

First of all, we load the *fRNC* package and the required data sets. The datasets contain 852 cancer samples and 18 normal samples, simultaneously measured all mRNA, miRNA, and protein expression. To save running time, we processed in advance to perform differential expression analysis and store internal data.

>library(fRNC)

>data("brca\_miRNA\_re\_se")

> data("brca\_RNA\_re\_se")

> data("brca\_pro\_re\_se")

>RNA\_re\_se\_type <- RNA\_re\_se$DEGs[,c("genes","logFC","PValue","type")]

>RNA\_re\_se\_type\_no\_RBP <- RNA\_re\_se\_type[which(RNA\_re\_se\_type$type != "rbp"),]

>dataN\_age\_all <- rbind(pro\_re\_se$DEGs[,c("genes","logFC","PValue","type")], RNA\_re\_se\_type\_no\_RBP, miRNA\_re\_se$DEGs[,c("genes","logFC","PValue","type")])

>gene2weight\_age <- combinp(dataN\_age\_all[,c("type","logFC","PValue")], islog = T)

>interac\_RBP\_age <- interStringency(type = "Protein", spec ="hg",stringency = "high")

>interac\_RBP\_age <- interac\_RBP\_age[,c("node\_gene\_ID","type","target\_gene\_ID")]

Based on the *P* value generated from differential analysis, we used the greedy algorithm to calculate the scoring sub-networks of the seed in each node with the method of "local". By the parameter *seletN*, users can assign the node set as the seeds with ensemble ID. For example the " ENSG00000136450" is ensemble ID for SRSF1. The size of the resultant RNC can also be set with 5-15 by the parameter of "node" in the main function *runmodule*.

>res.list\_local <- runmodule(network = interac\_RBP\_age, gene2weight, method = "local", maxsize=15, seletN = c("ENSG00000136450") )

>names(res.list\_local)

"GNCW" "module" "module.score.matrix"

#GNCW, an object of igraph class, is the node‐weighted ncRNA network.

#module, an object of list, contains all the valid modules, the name of each record is the seed  
gene.

#module.score.matrix an object of matrix, contains data for the seed module: Zm (module score),

Zn (normalized module score).

Finally, the output is loaded and plotted with the following commands. The local sub-network of " ENSG00000136450"( SRSF1) can also be saved as XGMML file for further analysis.

>list\_local <- res.list\_local$module

>savelocalM(res.list\_local)

The following figure is the resulted RNC containing one RNA binding protein “SRSF1” in BRCA. Users can observe it in the Cytoscape environment. The miRNA node is represented by black square and RBP node by blue triangle.

temp_srsf1.tif

**Figure 2. The "local" RNC starting from** SR**SF1 for RBP-ncRNA network in BRCA**