**Tutorial for RSNET**

1. **Introduction**

RSNET is a MATLAB program for inferring Gene Regulatory Networks (GRN) from gene expression dataset based on redundancy silencing and network enhancement technique. In RSNET algorithm, highly dependent nodes are constrained in the model as network enhancement items to enhance real intercations and dimension of putative causality is reduced adaptively to remove weak and indirect connections.

1. **Matlab Code description**
   1. **RSNET** [net,net\_value,sig]=RSNET(y,X,lamda,alpha,gama,beta,t)

**2.1.1 Input**

|  |  |
| --- | --- |
| Code | Description |
| y | Expression of target gene |
| X | Expression of regulatory genes |
| lamda | Regularization parameter for optimization |
| alpha | Parameter for filtering genes with low correlations |
| gama | Parameter for constraint genes with high correlations |
| beta | Parameter for filtering noise genes. |
| t | Parameter for the integration of linear information |

* + 1. **Output**

|  |  |
| --- | --- |
| Code | Description |
| net | 0-1 network or graph inferred by RSNET |
| net\_value | Network with interaction strengthens |
| sig | Significance of each edge in the inferred network |

* 1. **Example\_RSNET.m**

As an example, ‘Example\_RSNET.m’ is to infer gene regulatory network from gene expression dataset.

% An example for using RSNET to infer GRN: inferring GRN for apple fruit develoopment

% from gene expression data.

% Version data:Feb.,2021

clear;clc;

%% Dataset input

filename = 'apple\_fruit\_development\_data.txt';

data = importdata(filename);

data\_gene\_expression = data.data; % size(data\_gene\_expression);

data\_gene\_name =data.textdata; % size(data\_gene\_name);

sample\_name = data\_gene\_name(1);

data\_gene\_name(1)= []; %data\_gene\_name(1:5); % size(data\_gene\_name)

%% Filter the genes with low varaince.

threshold = 5;

[data\_gene\_name\_diff,data\_gene\_expression\_diff] = WeakVarianceDelete(data\_gene\_name,data\_gene\_expression,threshold);

% size(data\_gene\_expression\_diff)

% size(data\_gene\_name\_diff)

Y =log2(data\_gene\_expression\_diff);

fprintf('Data prepared for %d genes! \n',size(Y,1));

%% Run RSNET method on the data

lamda = 1;

alpha = 0.1; % parameter for correlation

gama = 0.5; % parameter for prior information

beta = 0.1; % parameter for deleting the noise

t = 0.5; % Parameter for the interation of MI and RO; t:[0,1]

J\_na = zeros(size(Y,1),size(Y,1)); J\_s=J\_na;

% n\_gene = size(Y,1)

n\_gene = 10

% for i=1:size(Y,1)

for i=1:n\_gene % Chose few genes for running time

if mod(i,50)==0

fprintf('Network inferring for gene: i=%d of %d.\n',i,size(Y,1));

end

y = Y(i,:);

X = [Y(1:i-1,:);Y(i+1:size(Y,1),:)];

[net,net\_value]=RSNET(y',X',lamda,alpha,gama, beta,t) ;

J\_s(i,1:i-1) = net(1:i-1); J\_s(i,i+1:size(Y,1))=net(i:end);

J\_na(i,1:i-1) = net\_value(1:i-1); J\_na(i,i+1:size(Y,1))=net\_value(i:end);

end

Gval=J\_na; Gval=abs(Gval); G = Gval;

q=0.5; G(G<q) = 0;

for i=1:size(G,1)

for j=1:size(G,2)

if G(i,j)>=G(j,i)

G(j,i)=0;

end

end

end

% sum(G>0,1);sum(G>0,2); sum(sum(G>0));

% index=find(sum(G>0,2)>100);

% gene\_list = data\_gene\_name\_diff;

% gene\_list(index)

%% output a network for cytoscape

% turn the network from matrix to column with TF|gene|edge(0 or 1)

gene\_list = data\_gene\_name\_diff;

threshold = 0.5;

[testfile]=Connect\_for\_cytoscape\_threshold(threshold,G,gene\_list,gene\_list) ;

network\_size=size(testfile,1)

fprintf('NOTICE:\nThe Size of the Inferred Network is %d.\n',network\_size);

xlswrite('result\_network',testfile);

**3. Contact information**

If you encounter any problem, please do not hesitate to contact us at zhangxj@wbgcas.cn