

Article

## A method for identifying hierarchical sub-networks / modules and weighting network links based on their similarity in sub-network / module affiliation

**WenJun Zhang**

School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong

E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 6 August 2015; Accepted 28 September 2015; Published online 1 June 2016



### Abstract

Some networks, including biological networks, consist of hierarchical sub-networks / modules. Based on my previous study, in present study a method for both identifying hierarchical sub-networks / modules and weighting network links is proposed. It is based on the cluster analysis in which between-node similarity in sets of adjacency nodes is used. Two matrices, linkWeightMat and linkClusterIDs, are achieved by using the algorithm. Two links with both the same weight in linkWeightMat and the same cluster ID in linkClusterIDs belong to the same sub-network / module. Two links with the same weight in linkWeightMat but different cluster IDs in linkClusterIDs belong to two sub-networks / modules at the same hierarchical level. However, a link with a unique cluster ID in linkClusterIDs does not belong to any sub-networks / modules. A sub-network / module of the greater weight is the more connected sub-network / modules. Matlab codes of the algorithm are presented.

**Keywords** sub-networks; modules; hierarchy; identification; link weighting; cluster analysis.

**Network Pharmacology**

ISSN 2415-1084

URL: <http://www.iaees.org/publications/journals/np/online-version.asp>

RSS: <http://www.iaees.org/publications/journals/np/rss.xml>

E-mail: [networkpharmacology@iaees.org](mailto:networkpharmacology@iaees.org)

Editor-in-Chief: WenJun Zhang

Publisher: International Academy of Ecology and Environmental Sciences

### 1 Introduction

Some networks (including some biological networks) consist of sub-networks / modules. In some cases we need to identify hierarchical sub-networks / modules and weight links according to their similarity in affiliation of sub-networks / modules. Based on the method of Zhang and Li (2016), the present study proposes a method for identifying hierarchical sub-networks / modules and weighting network links. Matlab codes of the method are presented for use.

### 2 Method

Assume the adjacency matrix of a network  $X$  with  $m$  nodes (Zhang, 2012, 2016) is  $a=(a_{ij})_{m \times m}$ , where  $a_{ij}=1$ , if two nodes  $v_i$  and  $v_j$  are adjacent;  $a_{ij}=0$ , if  $v_i$  and  $v_j$  are not adjacent,  $a_{ij}=a_{ji}$ , and  $a_{ii}=0$ ;  $i, j=1,2,\dots, m$ . A

sub-network / module is a node set, and links within the sub-network / module are much more than links towards outside the sub-network / module.

Suppose we do not know sub-networks / modules in advance, and they need to be determined by cluster analysis. The present method, based on Zhang and Li (2016) is

(1) Define between-node similarity as follows

$$\begin{aligned} r_{ij} &= \sum_{k=1}^m (a_{ik} a_{jk}) / m & a_{ij} &= 1 \\ r_{ij} &= 0 & a_{ij} &= 0 \end{aligned} \quad i, j = 1, 2, \dots, m; i \neq j$$

where  $0 \leq r_{ij} \leq 1$ .  $r_{ij} = 1$ , means that the sets of adjacent nodes of two connected nodes,  $i$  and  $j$ , are the same;  $r_{ij} = 0$ , means that two nodes,  $i$  and  $j$ , are unconnected.  $r_{ij} = r_{ji}$ ,  $i, j = 1, 2, \dots, m$ . Between-node distance is then defined as

$$d_{ij} = 1 - r_{ij}$$

where  $0 \leq d_{ij} \leq 1$ .

At the start,  $m$  nodes are naturally  $m$  sub-networks / modules respectively.

(2) Calculate between sub-network / module distance. Suppose there are two sub-networks / modules,  $X$  and  $Y$ . The distance between  $X$  and  $Y$ , based on the longest distance between two nodes, is

$$d_{XY} = \max d_{ij}, \quad i \in X, j \in Y$$

(3) In the sets of clusters (sub-networks / modules), choose two clusters,  $A$  and  $B$ , with the minimum  $d_{XY}$  to combine into a new cluster (new sub-network / module)

$$d_{AB} = \min d_{XY}$$

(4) Return (2) to repeat the cluster procedure, until all of  $m$  nodes are eventually clustered into a cluster (network). If the minimum  $d_{AB}$  for two or more cluster procedures are the same, only the last cluster procedure in these cluster procedures is chosen for use.

(5) For each cluster in the same cluster procedure, weight all links in the cluster with the same weight,  $r_{AB}$  (i.e.,  $1 - d_{AB}$ ), and mark different clusters with different IDs, until all clusters and cluster procedures are traversed. To avoid weight 0 for singular links, let  $r_{AB} = 10^{-10}$ , if  $r_{AB} = 0$ .

Finally, two matrices, linkWeightMat and linkClusterIDs, are achieved. Two links with both the same weight in linkWeightMat and the same cluster ID in linkClusterIDs belong to the same sub-network / module. Two links with the same weight in linkWeightMat but different cluster IDs in linkClusterIDs belong to two sub-networks / modules at the same hierarchical level. However, a link with a unique cluster ID in linkClusterIDs does not belong to any sub-networks / modules. A sub-network / module of the greater weight is the more connected sub-network / module.

The following are Matlab codes of the algorithm (linkWeight.m)

```
%Reference: Zhang WJ. 2016. A method for identifying hierarchical sub-networks / modules and weighting network links based on their similarity in sub-network / module affiliation. Network Pharmacology, 1(2):
```

```
clear
```

```
str=input('Input the file name of adjacency matrix a (e.g., raw.txt, raw.xls, etc. Adjacency matrix is D=(dij)m*m, where m is the
```

```

number of nodes in the network. dij=1, if vi and vj are adjacent, and dij=0, if vi and vj are not adjacent; dii=0; i, j=1,2,..., m);
','s');
a=load(str);
dim=size(a); m=dim(1);
for i=1:m-1
for j=i+1:m
if (a(i,j)~=0)
r(i,j)=0;
for k=1:m
if ((a(i,k)==a(k,j)) & (a(i,k)~=0)) r(i,j)=r(i,j)+1; end
end
r(i,j)=r(i,j)/m; r(j,i)=r(i,j);
d(i,j)=1-r(i,j); d(j,i)=d(i,j);
else r(i,j)=0; r(j,i)=0; d(i,j)=1; d(j,i)=1;
end; end; end
d1=d; bb1=1;
u(bb1)=0; nu(bb1)=m;
for i=1:nu(bb1) x(bb1,i)=i; end
for i=1:nu(bb1) y(bb1,i)=1; end
while (nu(bb1)>1)
aa=1e+10;
for i=1:nu(bb1)-1
for j=i+1:nu(bb1)
if (d(i,j)<=aa) aa=d(i,j); end
end; end
aa1=0;
for i=1:nu(bb1)-1
for j=i+1:nu(bb1)
if (abs(d(i,j)-aa)<=1e-06)
aa1=aa1+1; v(aa1)=i; w(aa1)=j;
end; end; end
for i=1:nu(bb1) s(i)=0; end
nn1=0;
for i=1:aa1
if ((v(i)~=0) & (w(i)~=0))
nn1=nn1+1;
for j=1:aa1
if ((v(j)==v(i)) | (v(j)==w(i)) | (w(j)==w(i)) | (w(j)==v(i)))
s(v(j))=nn1; s(w(j))=nn1;
if (j~=i) v(j)=0; w(j)=0; end; end
end
v(i)=0; w(i)=0;
end; end
for i=1:nn1
y(bb1+1,i)=0;

```

```

for j=1:nu(bb1)
if (s(j)==i)
for k=1:m
if (x(bb1,k)==j) x(bb1+1,k)=i; end
end
y(bb1+1,i)=y(bb1+1,i)+y(bb1,j);
end; end; end
for i=1:nu(bb1)
if (s(i)==0)
nn1=nn1+1;
for k=1:m
if (x(bb1,k)==i) x(bb1+1,k)=nn1; end
end
y(bb1+1,nn1)=y(bb1,i); end
end;
bb1=bb1+1; u(bb1)=aa; nu(bb1)=nn1;
for i=1:nu(bb1)-1
for j=i+1:nu(bb1)
d(i,j)=-1e+10;
for k=1:m
if (x(bb1,k)==i)
for kk=1:m
if (x(bb1,kk)==j)
if (d1(k,kk)>d(i,j)) d(i,j)=d1(k,kk); end
end; end; end; end
d(j,i)=d(i,j);
end; end; end
for k=1:m
y(bb1,k)=1; end
for i=bb1-1:-1:1
rr=0;
for j=1:nu(i+1)
ww=0;
for k=1:m
if (y(i+1,k)==j) ww=ww+1; v(ww)=k; end
end
vv=0;
for ii=1:ww
ee=0;
for jj=ii-1:-1:1
if (x(i,v(ii))==x(i,v(jj))) y(i,v(ii))=y(i,v(jj)); break; end
ee=ee+1;
end
if (ee==ii-1) vv=vv+1; y(i,v(ii))=rr+vv; end
end

```

```

rr=rr+vv;
end; end
for k=1:bb1
rs(k)=1-u(k);
end;
s=1; i=0;
while (m>0)
ss=1;
for j=s+1:bb1
if (rs(j)==rs(s)) ss=ss+1; end;
end
s=s+ss; i=i+1; la(i)=s-1;
if (s>=bb1) break; end
end
bb1=i;
yy=zeros(m);
for k=1:bb1
for i=1:nu(la(k))
for j=1:m
if (y(la(k),j)==i) yy(k,j)=i; end;
end; end; end
for k=1:bb1
rss(k)=rs(la(k)); uu(k)=u(la(k)); nuu(k)=nu(la(k));
end
linkWeightMat=zeros(m);
linkClusterIDs=zeros(m);
id=0;
for k=1:bb1
for i=1:nuu(k)
numm=0;
for j=1:m
if (yy(k,j)==i) numm=numm+1; temp(numm)=j; end
end
sim=max(rss(k),1e-10);
id=id+1;
for ii=1:numm-1
for jj=ii+1:numm
if ((a(temp(ii),temp(jj))~0) & (linkWeightMat(temp(ii),temp(jj))==0))
linkWeightMat(temp(ii),temp(jj))=sim; linkWeightMat(temp(jj),temp(ii))=sim;
linkClusterIDs(temp(ii),temp(jj))=id; linkClusterIDs(temp(jj),temp(ii))=id;
end
end; end; end; end
dif=zeros(1,m);
su=0;
for i=1:m-1

```

```

for j=i+1:m
sm=0;
for k=1:su
if ((linkClusterIDs(i,j)~=0) & (linkClusterIDs(i,j)~=dif(k)))
sm=sm+1;
end; end
if (sm==su) su=su+1; dif(su)=linkClusterIDs(i,j); end
end; end
ma=max(max(linkClusterIDs));
for k=1:su
for i=1:m-1
for j=i+1:m
if ((linkClusterIDs(i,j)~=0) & (linkClusterIDs(i,j)==dif(k)) & (linkClusterIDs(i,j)~=ma))
linkClusterIDs(i,j)=k; linkClusterIDs(j,i)=k;
end; end; end; end
idnew=su;
for i=1:m-1
for j=i+1:m
if (linkClusterIDs(i,j)==ma)
idnew=idnew+1; linkClusterIDs(i,j)=idnew; linkClusterIDs(j,i)=idnew;
end; end; end
for k=1:idnew-1
s=0;
for i=1:m-1
for j=i+1:m
if (k~=linkClusterIDs(i,j)) s=s+1; end
end; end
if (s==(m*(m-1)/2))
for i=1:m-1
for j=i+1:m
if (linkClusterIDs(i,j)>k) linkClusterIDs(i,j)=linkClusterIDs(i,j)-1; linkClusterIDs(j,i)=linkClusterIDs(i,j); end;
end; end;
end; end
lab=zeros(1,idnew-1);
for k=1:idnew-1
s=0;
for i=1:m-1
for j=i+1:m
if (k==linkClusterIDs(i,j)) s=s+1; end
end; end;
lab(k)=s;
end
iss="";
for k=1:idnew-1
la=0;

```

```
for i=1:m-1
for j=i+1:m
if (linkClusterIDs(i,j)==k) weig=linkWeightMat(i,j); la=1; break; end;
end;
if (la==1) break; end
end;
if (lab(k)>1) iss=strcat(iss,'\nLinks in sub-network / module (cluster) ID:',num2str(k),'(Link weight=',num2str(weig),' )\n'); end;
if (lab(k)==1) iss=strcat(iss,'\nStandalone link ID:',num2str(k),'(Link weight=',num2str(weig),' )\n'); end;
for i=1:m-1
for j=i+1:m
if (k==linkClusterIDs(i,j)) iss=strcat(iss,'\n',num2str(i),' ',num2str(j),' '); end
end; end;
end
fprintf(iss)
fprintf('\n\nLink weights\n')
linkWeightMat
fprintf('\nLink cluster IDs\n')
linkClusterIDs
```

### 3 Example and Application

#### 3.1 A typical example

The adjacency matrix (40×40) of the network used for typical case is

```
0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0

```

Running the algorithm, the links in each of sub-network (cluster) are as follows, where “Empty” means the cluster ID does not exist in the link cluster IDs matrix, linkClusterIDs

Links in sub-network / module (cluster) ID:1(Link weight=0.1)

(1,2)(1,3)(1,4)(1,5)(1,6)(2,3)(2,4)(2,5)(2,6)(3,4)(3,5)(3,6)(4,5)(4,6)(5,6)

Links in sub-network / module (cluster) ID:2(Link weight=0.05)

(10,11)(10,12)(10,13)(11,12)(11,13)(12,13)

Links in sub-network / module (cluster) ID:3(Link weight=0.05)

(15,16)(15,17)(15,18)(16,17)(16,18)(17,18)

Links in sub-network / module (cluster) ID:4(Link weight=0.15)

(24,25)(24,26)(24,27)(24,28)(24,29)(24,30)(24,31)(25,26)(25,27)(25,28)(25,29)(25,30)(25,31)(26,27)(26,28)(26,29)(26,30)(26,31)(27,28)(27,29)(27,30)(27,31)(28,29)(28,30)(28,31)(29,30)(29,31)(30,31)

Links in sub-network / module (cluster) ID:5(Link weight=0.1)

(35,36)(35,37)(35,38)(35,39)(35,40)(36,37)(36,38)(36,39)(36,40)(37,38)(37,39)(37,40)(38,39)(38,40)(39,40)

Standalone link ID:6(Link weight=1e-010)

(6,7)

Standalone link ID:7(Link weight=1e-010)

(7,8)

Standalone link ID:8(Link weight=1e-010)

(8,9)

Standalone link ID:9(Link weight=1e-010)

(9,10)

Standalone link ID:10(Link weight=1e-010)

(13,14)

Standalone link ID:11(Link weight=1e-010)

(14,15)

Standalone link ID:12(Link weight=1e-010)

(18,19)

Standalone link ID:13(Link weight=1e-010)

(19,20)

Standalone link ID:14(Link weight=1e-010)

(20,21)

Standalone link ID:15(Link weight=1e-010)

(21,22)







matrices.

### 3.2 Application in a phylogenic network

The adjacency matrix of phylogenic network for 17 common HLA-DQB1 alleles of the world's 12 human races and populations is from Zhang and Qi (2014) and Zhang (2015, supplementary material, HLA\_DQB1\_adj.txt). Use the algorithm above, the resultant link weights matrix, linkWeightMat, is

0	0	0	0	0	0	*	0	0	0	0	0
0	0	*	*	0	0	*	*	0	0	0	0
0	*	0	0.5	0.4167	0.4167	0.5	0.5	0	0.4167	0	0
0	*	0.5	0	0.4167	0.4167	0.5	0.5	0	0.4167	0	0
0	0	0.4167	0.4167	0	0.5	0.4167	0.4167	0	0.4167	*	0
0	0	0.4167	0.4167	0.5	0	0.4167	0.4167	0	0.4167	*	0
*	*	0.5	0.5	0.4167	0.4167	0	0.5	0	0.4167	0	0
0	*	0.5	0.5	0.4167	0.4167	0.5	0	*	0.4167	0	0
0	0	0	0	0	0	0	*	0	0	0	0
0	0	0.4167	0.4167	0.4167	0.4167	0.4167	0.4167	0	0	0	0
0	0	0	0	*	*	0	0	0	0	0	*
0	0	0	0	0	0	0	0	0	0	*	0

“\*” denotes  $10^{-10}$

The following is the link cluster IDs matrix, linkClusterIDs

0	0	0	0	0	0	5	0	0	0	0	0
0	0	6	7	0	0	8	9	0	0	0	0
0	6	0	2	3	3	2	2	0	3	0	0
0	7	2	0	3	3	2	2	0	3	0	0
0	0	3	3	0	4	3	3	0	3	10	0
0	0	3	3	4	0	3	3	0	3	11	0
5	8	2	2	3	3	0	2	0	3	0	0
0	9	2	2	3	3	2	0	12	3	0	0
0	0	0	0	0	0	0	12	0	0	0	0
0	0	3	3	3	3	3	3	0	0	0	0
0	0	0	0	10	11	0	0	0	0	0	13
0	0	0	0	0	0	0	0	0	0	13	0

## 4 Discussion

In addition to the method of link-weighting in present study, we can directly use the link cluster IDs matrix, linkClusterIDs, to weight links according to different cluster IDs in the matrix. In this case, the link-weighting can even be conducted in Excel, Notepad, etc., for small networks. The Matlab codes following the above codes are thus

```
str=input('Input the file name of sub-network / module based link-weighting data (there are idnew-1 rows, each row has two values, i.e. cluster ID and link weight; where idnew-1 is the total number of cluster IDs): ','s');
```

```
wei=load(str);
for k=1:idnew-1
for i=1:m-1
for j=i+1:m
if (linkClusterIDs(i,j)==k) linkWeightMat(i,j)=wei(k,2); linkWeightMat(j,i)=wei(k,2);end
end; end; end
fprintf('\n\nLink weights\n')
linkWeightMat
```

The algorithm is particularly suggested for using in the situations that there are some remarkable sub-networks / modules in the network. Furthermore, it is also applicable to other networks without significant sub-networks / modules.

### Acknowledgment

I am thankful to the support of Discovery and Crucial Node Analysis of Important Biological and Social Networks (2015.6-2020.6), from Yangling Institute of Modern Agricultural Standardization, High-Quality Textbook *Network Biology* Project for Engineering of Teaching Quality and Teaching Reform of Undergraduate Universities of Guangdong Province (2015.6-2018.6), from Department of Education of Guangdong Province, and Project on Undergraduate Teaching Reform (2015.7-2017.7), from Sun Yat-sen University, China.

### References

- Zhang WJ. 2012. Computational Ecology: Graphs, Networks and Agent-based Modeling. World Scientific, Singapore
- Zhang WJ. 2015. Prediction of missing connections in the network: A node-similarity based algorithm. *Selforganizology*, 2(4): 91-101
- Zhang WJ. 2016. *Selforganizology: The Science of Self-Organization*. World Scientific, Singapore
- Zhang WJ, Li X. 2016. A cluster method for finding node sets / sub-networks based on between-node similarity in sets of adjacency nodes. *Proceedings of the International Academy of Ecology and Environmental Sciences*, 6(1): 13-23
- Zhang WJ, Qi YH. 2014. Pattern classification of HLA-DRB1 alleles, human races and populations: Application of self-organizing competitive neural network. *Selforganizology*, 1(3-4): 138-142