Screening node attributes that significantly influence node centrality in the network

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 19 December 2015; Accepted 16 January 2016; Published online 1 September 2016

Abstract
In present study, I proposed a method to screen node attributes that significantly influencing node centrality in the network. One of four node centralities, degree centrality, closeness centrality, betweenness centrality, and circuit centrality, can be used as the dependent variable and attribute-by-node data are used as the data of independent variables. Stepwise linear regression method was applied to screen statistically significant node attributes from candidate attributes. Matlab codes of the algorithm are provided also.

Keywords network; node attributes; node centrality; attribute screening; stepwise linear regression.

1 Introduction
In nature, many networks are created by connections between nodes that have some correlations, similarities or complementarities in node attributes (Zhang, 2011, 2012a, 2012b, 2015a, 2015c). They include, for example, networks of protein-protein interactions, ecological networks, some social networks, etc. Node attributes determine the evolution and topological structure of such networks. However, in most cases, we do not know which attributes are significant or crucial. Utilization of insignificant attributes will increase the noise of network information. Thus it is necessary to screen significant node attributes from a lot of candidate attributes. In present study, I try to propose a method to screen node attributes that significantly influence node centrality in the network.

2 Algorithm
Suppose there is a weighted network $X$ with $m$ nodes (Zhang, 2012a), its weighted adjacency matrix is $d=(d_{ij})_{m \times m}$. $d_{ij}=w_{ij}$ if two nodes $v_i$ and $v_j$ are adjacent, and $d_{ij}=0$, if $v_i$ and $v_j$ are not adjacent, where $w_{ij}$ is the weight of the link from node $v_i$ to node $v_j$, $i, j=1, 2, \ldots, m$. For unweighted network, $w_{ij}=1$, if two nodes $v_i$ and $v_j$ are adjacent, $i, j=1, 2, \ldots, m$. Adjacency matrix $d$ is a symmetric matrix, i.e., $d=d^t$. Known $n$ candidate attributes (i.e., traits) for $m$ nodes (e.g., for $m$ proteins, the attributes can be pH, isoelectric point, molecular...
weight, or for \( m \) species, they can be \( n \) alleles, etc.). The raw data matrix is \( x = (x_{ij})_{n \times m} \).

First, calculate the following centrality indices of nodes, which represent one of the topological properties of a network (Shams and Khansari, 2014; Zhang, 2012a, 2012b, 2012c; Khansari et al., 2016).

(1) Degree centrality. It is a local centrality based on neighbourhood. It reflects the node influence on its neighbourhoods. A node degree centrality is the sum of the weights of the links attached to that node. It represents the whole involvement of a node in the network (Opsahl et al., 2010; Khansari et al., 2016)

\[
y_i = \sum_{j=1}^{n} w_{ij}
\]

where \( w_{ij} \) is the weight of link from node \( i \) to \( j \).

(2) Closeness centrality. It is a distance-based measure. For a distance-based measure, different values (cost, connection strength, etc.) can be assigned to different links in weighted networks. It is a global centrality which represents the independence of a node in the network (Freeman, 1978). Closeness centrality is defined as reciprocal of the sum of the node’s geodesic distances to all other nodes (the distances of the shortest paths) in the network

\[
y_i = 1/ \sum_{j=1}^{n} d_{ij}
\]

where, \( d_{ij} \) is the weighted geodesic path between node \( i \) to \( j \). Dijkstra algorithm (Dijkstra, 1959; Zhang, 2012a) is used to calculate the shortest path and corresponding distance between two nodes.

(3) Betweenness centrality. It is a distance-based measure. It represents the node’s ability to control the data flow in the network (Freeman, 1978). This measure is the proportion of number of geodesic paths that pass through the given node to total number of geodesic paths between any pair of nodes in the network (Khansari et al., 2016)

\[
y_j = n_i / s
\]

where \( n_i \) is the number of weighted geodesic paths which pass through node \( i \), and \( s \) is the total number of weighted geodesic paths between any pair of nodes in the network.

(4) Circuit centrality. Here I proposed circuit centrality to characterize the contribution of a node in different possible subgraphs. It is a distance-based measure and based on the concept of subgraph centrality (Khansari et al., 2016). This measure is the proportion of number of fundamental circuits (i.e., cycles, or closed paths, or loops, see Paton (1969) and Zhang (2012a)) that pass through the given node to total number of fundamental circuits in the network

\[
y_i = n_i / s
\]

where \( n_i \) is the number of fundamental circuits which pass through node \( i \), and \( s \) is the total number of fundamental circuits in the network.

In present study, I use the stepwise linear regression to screen node attributes. The stepwise linear regression is a multi-variable regression that can screen statistically significant variables into the linear regression equation (Zhang and Fang, 1982). The full multi-variable linear regression equation is (Qi et al., 2016)

\[
y=b_0+ b_1 x_1 + b_2 x_2 + ... + b_n x_n
\]
where $x_i$ is the $i$-th attribute. Let

$$
l_{ij} = \frac{\sum_{k=1}^{m} x_{ik} x_{jk}}{m} \quad l_{ji} = \frac{\sum_{k=1}^{m} x_{jk} x_{ik}}{m}
$$

Let

$$
l_{iy} = \frac{\sum_{k=1}^{m} x_{ik} y_k - \left( \frac{\sum_{k=1}^{m} x_{ik} \sum_{k=1}^{m} y_k}{m} \right)}{m}
$$

$i, j=1,2,\ldots, n$

Correlation coefficients between-attribute and between attribute and centrality index $y$ are

$$r_{ij} = \frac{l_{ij}(l_{ji})}{(l_{ii} l_{jj})^{0.5}}$$

$$r_{iy} = \frac{l_{iy}}{(l_{ii} l_{yy})^{0.5}}$$

Solve the equation

$$r_{i1} b_1' + r_{i2} b_2' + \ldots + r_{in} b_n' = r_{iy}
$$

$i=1,2,\ldots, n$

The variance contribution of each attribute is

$$v_i = \frac{r_{iy}^2}{r_{ii}^2}$$

Let $v_i = \max v_i$, and calculate $F=(m-l-1)v_i/q$, where $l$ is the number of attributes screened into the equation, $q$ is the square of residuals. For first screening, $q=v_k$. If $F \geq F_a$, screen the attribute $x_k$ into the equation ($F_a=0.1$, etc.), or else remove $x_k$. The correlation matrix are changed as the following

$$r_{ij} = r_{ij} - \frac{r_{ik} r_{jk}}{r_{kk}} \quad i \neq k$$

$$r_{ik} = r_{ik} / r_{kk} \quad j \neq k, i=k$$

$$r_{kk} = 1 / r_{kk} \quad i=k, j=k$$

where $k$ is the $k$-th screened or removed attribute. Calculate $v_k(l+1)=\max v_i(l+1)$, and $F=(m-l-2) v_k(l+1)/(q(l)-v_k(l+1))$. If $F(l+1) \geq F_a$, screen the attribute $x_k$ into the equation, and change the correlation matrix. Let $v_k = \max v_i$, where $x_k$ is the attribute already in the equation, $F_k=(m-l-1) v_k(l)/q(l)$, where $q$ is the $r_{yy}$ in the inverse matrix of correlation matrix. If $F_k \leq F_a$, remove the attribute $x_k$ from the equation, otherwise screen into the attribute. Repeat the procedure above, until no attribute can be screened into or remove from the equation.

By doing so, the linear regression equation is obtained as the following

$$y = \tilde{b}_0 + \tilde{b}_1 x_i + \ldots + \tilde{b}_j x_j + \ldots + \tilde{b}_k x_k$$

and the attributes remained in the equation are qualified node attributes.

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

%Reference: Zhang WJ. 2016. Screening node attributes that significantly influence node centrality in the network.
Selforganizology, 3(3): 75-86

raw=input('Input the file name of node-by-attribute data (e.g., raw.txt, raw.xls, etc. The matrix is z=(zij)n*m, where n is the number of candidate attributes, m is total number of nodes): ','s');
adj=input('Input the file name of adjacency matrix of unweighted or weighted network (e.g., adj.txt, adj.xls, etc. Adjacency matrix is d=(dij)m*m, where m is the number of nodes in the network. dij=1 for unweighted network and dij=wij for weighted network): ','s');
network (wij is the weight for the link vi to vj), if vi and vj are adjacent, and dij=0, if vi and vj are not adjacent; i, j=1,2,…, m):
's');
fs=input('Input the F threshold value for screening attributes (e.g., 0.1, 0.05): ');
choice=input('Input the type of topological property of nodes (1: Degree centrality; 2: Closeness centrality; 3: Betweenness centrality): ');
raw=load(raw); nw=size(raw,1); m=size(raw,2); n=nw+1;
x=zeros(n,m);
adj=load(adj);
newdata=zeros(nw,m); xb=zeros(1,n); sg=zeros(1,n); ds=zeros(1,n); degr=zeros(1,m);
switch choice
    case 1
degr=sum(adj);
    case 2
[ss,pat,distances,paths]=Dijkstra(adj);
    for i=1:m
degr(i)=1/sum(distances(i,:));
    end
    case 3
[ss,pat,distances,paths]=Dijkstra(adj);
    for i=1:m
degr(i)=pat(i)/ss;
    end
end
a=zeros(n);
xb=[raw;degr];
iss=';for i=1:n
c=0;
for j=1:m
c=c+x(i,j);
end
xb(i)=c/m;
c=0;
for j=1:m
c=c+(x(i,j)-xb(i))^2;
end
sg(i)=sqrt(c);
end
h=sg(n);
for i=1:n-1
    for j=i+1:n
c=0;
    for k=1:m
c=c+(x(i,k)-xb(i))*(x(j,k)-xb(j));
    end
    end
end

IAEES
\[ a(i,j) = \frac{c}{(sg(i) * sg(j))}; \quad a(j,i) = a(i,j); \]
end; end
for \( i = 1:n \)
\[ xb(i) = i; \quad sg(i) = 0; \quad a(i,i) = 1; \]
end
\[ l = 0; \quad s = 0; \]
while \((n > 1)\)
if \((l = n-1)\) break; end
\[ ma = 0; \]
for \( i = 1:n \)
\[ ds(i) = xb(i); \]
end
for \( i = 1:n-1 \)
if \((ds(i) == 0)\) continue; end
if \((a(i,i) < 1e-05)\) continue; end
\[ v1 = a(i,n) * a(n,i) / a(i,i); \]
if \((v1 > ma)\) ma = v1; \( k = i; \) end
end
\[ f1 = ma * (m-l-2) / (a(n,n) - ma); \]
if \((f1 <= fs)\) break; end
\[ xb(k) = 0; \quad sg(k) = k; \]
\[ l = l + 1; \]
for \( i = 1:n \)
for \( j = 1:n \)
if \((i == k) \& (j == k)\) \[ a(i,j) = a(i,j) - a(i,k) * a(k,j) / a(k,k); \]
end; end
for \( j = 1:n \)
if \((j == k)\) \[ a(k,j) = a(k,j) / a(k,k); \]
\[ a(j,k) = -a(j,k) / a(k,k); \]
end
\[ a(k,k) = 1 / a(k,k); \]
\[ r = \sqrt{1 - a(n,n)}; \]
\[ yn = h * \sqrt{a(n,n) / (m-l-1)}; \]
if \((s == 0)\) s = 1; continue; end
\[ lab = 0; \]
while \((n > 1)\)
\[ ma = -1e+18; \]
for \( i = 1:n \)
\[ ds(i) = sg(i); \]
end
for \( i = 1:n-1 \)
if \((ds(i) == 0)\) continue; end
if \((a(i,i) < 1e-05)\) continue; end
\[ v1 = a(i,n) * a(n,i) / a(i,i); \]
if \((v1 > ma)\) ma = v1; \( k = i; \) end
end
f1=-ma*(m-l-1)/a(n,n);
if (f1>fs) lab=1; break; end
sg(k)=0; xb(k)=k; l=l-1;
for i=1:n
for j=1:n
if ((i==k) & (j==k)) a(i,j)=a(i,j)-a(i,k)*a(k,j)/a(k,k); end
end
for j=1:n
if (j==k) a(k,j)=a(k,j)/a(k,k); a(j,k)=-a(j,k)/a(k,k); end
end
a(k,k)=1/a(k,k);
end;
if (lab==1) continue; end
end
for i=1:n-1
a(i,1)=sg(i);
end
for i=1:n
c=0;
for j=1:m
c=c+x(i,j);
end
xb(i)=c/m;
c=0;
for j=1:m
c=c+(x(i,j)-xb(i))^2;
end
sg(i)=sqrt(c);
end
h=sg(n);
c=0;
for i=1:n-1
if (a(i,1)==0) continue; end
ds(i)=a(i,n)*sg(n)/sg(i); a(i,2)=ds(i); c=c+ds(i)*xb(i);
end
s=xb(n)-c;
iss=strcat(iss,'Screened attributes: 
');
nm=0;
for i=1:n-1
if (a(i,1)==0) continue; end
if (ds(i)==0) iss=strcat(iss,'Attribute-',num2str(i)); end
if ((ds(i+1)==0) & (i<n-1)) iss=strcat(iss,''); end
if (ds(i)==0)
nm=nm+1;
end
for j=1:m
    newdata(nm,j)=x(i,j);
end; end
end
fprintf(‘\nNew attribute-by-node data’)
disp([newdata(1:nm,:)])
iss=strcat(iss,’\nStepwise regression equation:‘);
iss=strcat(iss,’y=‘,num2str(s));
for i=1:n-1
    if (a(i,1)==0) continue; end
    if (ds(i)>0) e1=num2str(ds(i)); end
    if (ds(i)<0) e1=num2str(abs(ds(i))); end
    if (ds(i)>0) iss=strcat(iss,’+‘,e1,’Attribute’,num2str(i)); end
    if (ds(i)<0) iss=strcat(iss,’-‘,e1,’Attribute’,num2str(i)); end
end
iss=strcat(iss,’\nCorrelation coefficient R=‘,num2str(r),’, ‘,’F value=‘,num2str(fs),’,’n‘);
fprintf(iss)

The functions, Dijkstra.m, foundCircuit.m, which are used to calculate the shortest path and corresponding
distance between two nodes, and to calculate the fundamental circuits in the network respectively (Zhang,
2012a), are as the following

function [ss,pat,distances,paths]=Dijkstra(d)
    % d: weighted adjacency matrix; ss: total number of paths; pat: number of paths passing through each node; distances: matrix of
distances between different nodes; paths: string of paths and distances between any of two nodes
    v=size(d,1);
    p=zeros(1,v); w=zeros(1,v); a=zeros(1,v); b=zeros(1,v);
    pat=zeros(1,v);
    distances=zeros(v);
    for i=1:v
        for j=1:v
            if ((d(i,j)==0) & (i~=j)) d(i,j)=inf; end
        end
        paths="\n        su=0;
        for j=1:v-1
            for k=j+1:v
                for i=1:v
                    p(i)=0; w(i)=0;
                    a(i)=inf;
                    end
                a(j)=0; w(j)=1; n=j; h=0;
                while (h>0)
                    ma=inf;
                    for i=1:v
                        IAEES

www.iaees.org
if (w(i)==1) continue; end
iv=d(n,i)+a(n);
if (iv>a(i)) a(i)=iv; b(i)=n; end
if (a(i)>ma) continue; end
ma=a(i); h=i;
end
w(h)=1;
if (h==k) break; end
n=h;
end
ds=a(k); p(1)=k; c=k;
for i=2:v
if (c==j) break; end
p(i)=b(c); c=b(c);
end
paths=strcat(paths,'Shortest path from ',num2str(j),' to ',num2str(k),':
for i=v:-1:1
if ((p(i)==0) & (sd==inf))
if (i>1) paths=strcat(paths,num2str(p(i)),'->'); end
if (i<=1) paths=strcat(paths,num2str(p(i)),'
'); end
end; end
for i=v:-1:1
for h=1:v
if (p(i)==h) pat(h)=pat(h)+1; break; end
end; end
if (sd==inf) paths=strcat(paths,'No path','
'); su=su+1; end
if (sd==inf) paths=strcat(paths,'Distance=',num2str(sd),'
'); distances(j,k)=sd; distances(k,j)=sd; end
end; end
ss=v*(v-1)/2-su;

function [num,n,circuits]=foundCircuit(d)
% d: weighted adjacency matrix; num: total number of fundamental circuits; n: number of fundamental circuits containing each node; circuits: string of all circuits
v=size(d,1);
l=zeros(1,v); vp=zeros(1,v); ts=zeros(1,v); circuit=zeros(1,v*(v-1)/2);
n=zeros(1,v);
num=0;
for i=1:v
l(i)=-1;
end
circuits="n";
t=1;
while (v>0)
its=1;
hs(1)=t; l(t)=0;

IAEES
while (v>0)
if (its==0) break; end
r=ts(its); lm=l(r)+1;
for w=1:v
if (d(r,w)<=0) continue; end
if ((d(r,w)>0) & ((l(w)+1)==0))
ts(its)=w;
its=its+1;
vp(w)=r; l(w)=lm;
d(r,w)=0; d(w,r)=0;
continue; end
num=num+1; a=vp(w);
m=1;
circuit(1)=r; j=r;
while (v>0)
j=vp(j);
m=m+1;
circuit(m)=j;
if (j==a) break; end
end
m=m+1;
circuit(m)=w;
circuits=strcat(circuits,'Number of fundamental circuit: ',num2str(num),'
');
circuits=strcat(circuits,'Fundamental circuit: '); for j=1:m
circuits=strcat(circuits,num2str(circuit(j)),',->');
end
circuits=strcat(circuits,num2str(circuit(1)),',n');
for i=1:v
for j=1:m
if (circuit(j)==i) n(i)=n(i)+1; break; end
end
end
d(r,w)=0; d(w,r)=0;
end
its=its-1;
end
la=0;
for t=t:v
if ((l(t)==-1) la=1; break; end
end
if (la==1) continue; end
break;
end
3 Application Example

Data of 17 common HLA-DQB1 alleles (candidate attributes) for the world’s 12 human races and populations (nodes) (17×12; Table 1) are from Zhang and Qi (2014) and Zhang (2015c, supplementary material; here the transposition of HLA_DQB1.txt). In addition, an adjacency matrix (12×12; HLA_DQB1_adj.txt; Zhang, 2015c, supplementary material), derived from linear correlation analysis, is given.

Table 1 The world’s 12 human races and populations (nodes) and 17 common HLA-DQB1 alleles (candidate attributes)

<table>
<thead>
<tr>
<th>Allele ID</th>
<th>Alleles 1 2 3 4 5 6 7 8 9 10 11 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DQB1*0201</td>
</tr>
<tr>
<td>2</td>
<td>DQB1*0301</td>
</tr>
<tr>
<td>3</td>
<td>DQB1*0302</td>
</tr>
<tr>
<td>4</td>
<td>DQB1*0303</td>
</tr>
<tr>
<td>5</td>
<td>DQB1*0401</td>
</tr>
<tr>
<td>6</td>
<td>DQB1*0402</td>
</tr>
<tr>
<td>7</td>
<td>DQB1*0501</td>
</tr>
<tr>
<td>8</td>
<td>DQB1*0502</td>
</tr>
<tr>
<td>9</td>
<td>DQB1*05031</td>
</tr>
<tr>
<td>10</td>
<td>DQB1*05032</td>
</tr>
<tr>
<td>11</td>
<td>DQB1*0504</td>
</tr>
<tr>
<td>12</td>
<td>DQB1*0601</td>
</tr>
<tr>
<td>13</td>
<td>DQB1*0602</td>
</tr>
<tr>
<td>14</td>
<td>DQB1*0603</td>
</tr>
<tr>
<td>15</td>
<td>DQB1*0604</td>
</tr>
<tr>
<td>16</td>
<td>DQB1*06051</td>
</tr>
<tr>
<td>17</td>
<td>DQB1*null</td>
</tr>
</tbody>
</table>

Table 2 Screened node attributes when different node centralities (dependent variables) and statistic significance levels are used. ‘+’ denotes positive influence and ‘-’ denotes negative influence.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Node centrality</th>
<th>Closeness centrality</th>
<th>Betweenness centrality</th>
<th>Circuit centrality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F=0.1</td>
<td>F=0.05</td>
<td>F=0.1</td>
<td>F=0.05</td>
</tr>
<tr>
<td>1</td>
<td>DQB1*0201</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>DQB1*0301</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>DQB1*0302</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>DQB1*0303</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>DQB1*0401</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>DQB1*0402</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>7</td>
<td>DQB1*0501</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>DQB1*0502</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>DQB1*05031</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>DQB1*05032</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>11</td>
<td>DQB1*0504</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>12</td>
<td>DQB1*0601</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>13</td>
<td>DQB1*0602</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>14</td>
<td>DQB1*0603</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>15</td>
<td>DQB1*0604</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>DQB1*06051</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>DQB1*null</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

R 0.9999 0.9999 0.9999 0.9999 1 1 0.9998 0.9998
In Table 1, nodes 1 to 12 represent Tibetan, Uighur, Kazak, Xingjiang Han, Taiwanese, Hong Kong, Northern Han, Shanghai Han, Hunan Han, Manchu, Buyi, and Dai (Geng et al., 1995; Chang and Hawkins, 1997; Mizuki et al., 1997, 1998, and others).

Using different node centralities and statistic significances, screening results of node attributes are listed in Table 2. As an example, the stepwise linear regression equation for node centrality and $F=0.1$ is

$$y=-0.29311 + 0.00899 \times \text{Attribute1} - 0.00963 \times \text{Attribute3} + 0.03587 \times \text{Attribute4} - 0.01650 \times \text{Attribute8} - 0.25115 \times \text{Attribute10} + 0.10231 \times \text{Attribute11} + 0.01261 \times \text{Attribute12} + 0.05268 \times \text{Attribute14} - 0.01650 \times \text{Attribute15} + 0.01261 \times \text{Attribute12} + 0.05268 \times \text{Attribute14} - 0.01650 \times \text{Attribute15} + 0.09326 \times \text{Attribute17}$$

$$R=0.9999$$

4 Discussion
In present study, a single topological property, i.e., a type of node centrality is used as the dependent variable of stepwise regression, of which degree centrality is particularly important. Degree centrality is directly correlated to network structure, link prediction, and network evolution (Zhang, 2012a, 2015a, 2015b; Zhang and Li, 2015a, 2015b). To find node attributes that significantly determine the comprehensive topological structure of the network, multiple properties, e.g., node centralities, connectivity (Zhang, 2012a), connectance (2011, 2012a), aggregation index (Zhang and Zhan, 2011), etc., should be jointly used. In this case, the stepwise regression of multiple properties vs. multiple node attributes (Zhang and Fang, 1982) is suggested for use.

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
Chang YW, Hawkins BR. 1997. HLA Class I and Class II frequencies of a Hong Kong Chinese population based on bone marrow donor registry data. Human Immunology, 56: 125-135
Zhang WJ, Li X. 2015b. Linear correlation analysis in finding interactions: Half of predicted interactions are undeterministic and one-third of candidate direct interactions are missed. Selforganizology, 2(3): 39-45