Article

Network matrix based methods for between-network comparison

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 9 August 2017; *Accepted 15 October 2017*; *Published 1 December 2018*

Abstract
In present article, I introduced some network matrix based methods for comparing and testing between-network difference/similarity, including the methods for interval weights based network matrix, including between-network similarity, randomization test of between-network difference, and statistic test of between-network difference, and the method for Boolean weights based network matrix. In addition, degree change index, weight change index, and eigenvector matrix change index were presented also. Matlab codes of the methods were provided.

Keywords network matrix; network comparison; difference; similarity; algorithms; Matlab.

Network Biology
ISSN 2220-8879
URL: http://www.iaees.org/publications/journals/nb/online-version.asp
RSS: http://www.iaees.org/publications/journals/nb/rss.xml
E-mail: networkbiology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

The structure of network covers node degree, network connectance, aggregation strength, etc (Dormann,2011; Zhang and Zhan, 2011). Various statistics may be used in the difference comparison of network structure (Solow, 1993; Manly, 1997; Zhang, 2012a, 2018). For example, a Java algorithm has been presented to statistically compare between-network structure difference (Zhang, 2011a). In addition, the Java algorithm was also developed to statistically compare between-community structure difference (Zhang, 2011b). In present article, I introduce some network matrix based methods to compare and test between-network difference/similarity.

2 Methods

Suppose we compare the two networks *A* and *B*, based on their network matrices $A=(a_{ij})$ and $B=(b_{ij})$, i, j=1, 2, ..., m, where m is the number of nodes, a_{ij} and b_{ij} are the weights between two nodes i and j in two networks respectively.

2.1 Interval weights based network matrix

Assume the weights a_{ij} and b_{ij} are all interval values. Several methods bellow can be used to compare two networks. In these methods, network matrices *A* and *B* are firstly transformed to vectors A_1 and B_1 respectively, i.e., (A_1, B_1)={(a_{ij}, b_{ij}) | i, j=1, 2, ..., m}, and the corresponding methods are thus used.

2.1.1 Between-network similarity

By using Pearson correlation, etc (Zhang, 2012a, 2012c, 2015, 2018; Zhang and Li, 2015a, 2015b), we can achieve the test results for network similarity/correlation. Generalized Matlab codes for Pearson correlation between two network matrices, *A* and *B*, are as follows (Zhang, 2012a, 2018)

```
AA=input('Input the excel file name of network matrix A: ','s');
A=xlsread(AA);
BB=input('Input the excel file name of network matrix B: ','s');
B=xlsread(BB);
m=size(A,1); mm=size(A,2); n=size(B,1); nn=size(B,2);
if ~((m==mm) & (n==nn) & (m==n))
error('Network matrices A and B should be the square matrices of the same size.');
end
% Matrices A and B are the square matrices of the same size.
sig=input('Input significance level(e.g., 0.01): ');
A1=reshape(A, m*m, 1);    %Transform matrix A to vector A1
B1=reshape(B, m*m, 1);    %Transform matrix B to vector B1
r=corr(A1,B1);
fprintf(['Pearson correlation r=' num2str(r) '\n']);
tvalue=abs(r)/sqrt((1-r^2)/(m*m-2));
p=(1-tcdf(tvalue,m*m-2))*2;
sigma=p<sig;
if (sigma==1) fprintf(['Pearson correlation is statistically significant (p=' num2str(p) ')\n']); end
if (sigma==0) fprintf(['Pearson correlation is not statistically significant (p=' num2str(p) ')\n']); end
```

2.1.2 Randomization test of between-network difference

By using randomization test for between-network difference based on Euclidean distance, Manhattan distance, Chebyshov distance, and Pearson correlation-based distance (Zhang, 2012a, 2015, 2018; Manly, 1997; Schoenly and Zhang, 1999; Solow, 1993; Zhang and Schoenly, 1999), we can achieve the test results for network difference/similarity. Generalized Matlab codes for randomization test between two network matrices, *A* and *B*, are as follows (Zhang, 2012a, 2018)

```
AA=input('Input the excel file name of network matrix A: ','s');
A=xlsread(AA);
BB=input('Input the excel file name of network matrix B: ','s');
B=xlsread(BB);
m=size(A,1); mm=size(A,2); n=size(B,1); nn=size(B,2);
if ~((m==mm) & (n==nn) & (m==n))
error('Network matrices A and B should be the square matrices of the same size.');
end
% Matrices A and B are the square matrices of the same size.
sim=input('Input the maximum number of simulations (e.g., 100): ');
sel=input('Choose distance measure (1: Euclidean distance; 2: Manhattan distance; 3: Chebyshov distance; 4: Pearson
correlation): ');
```

```
sig=input('Input the significance level (e.g., 0.05): ');
A1=reshape(A, m*m, 1);    %Transform matrix A to vector A1
B1=reshape(B, m*m, 1);    %Transform matrix B to vector B1
pvalue=randTest(A1,B1,sim,sel);
if (pvalue<sig) fprintf(['Difference is statistically significant (p=' num2str(pvalue) ')\n']); end
if (pvalue>=sig) fprintf(['Difference is not statistically significant (p=' num2str(pvalue) ')\n']); end
```

The functions, randTest.m, euclideandis.m, manhattandis.m, chebyshovdis.m, and correcoeffdiff.m, used in the algorithm above are as follows (Zhang, 2018)

```
functionpvalue=randTest(x,y,sim,sel)            %pvalue: calculated p value.
if ((min(size(x))~=1) | (min(size(y))~=1))      %sim: times of randomizations.
error('Both x and y are vectors');
end                                             %x and y: two vectors to be tested. x and y are row vectors.
m=max(size(x));
if (max(size(y))~=m)
error('Vector sizes do not match.');
end
if (sim<=1)
error('No. randomizations are too less.');
end
dum=min(min(x),min(y));
if (dum<0)
x=x-dum;
y=y-dum;
end
ma=-1e10;
for j=1:2
for i=1:m
in=1;
if (j==1) dum=x(i);
else dum=y(i);
end
while (m~=0)
if ((abs(dum-floor(dum))<1) & (~(abs(dum-floor(dum))<=1e-10)))
in=in*10;
dum=dum*10;
if ((floor(dum+1e-10))~=(floor(dum))) break; end
else break; end
end
if (in>ma) ma=in; end
end; end
x=x.*ma.*1.0;
y=y.*ma.*1.0;
switch sel
```

```
case 1
dxy=euclideandis(x,y);
case 2
dxy=manhattandis(x,y);
case 3
dxy=chebyshovdis(x,y);
case 4
dxy=correcoeffdiff(x,y);
end
nrx=sum(x);
nrxy=sum(sum(x+y));
fr=0;
for sm=1:sim
ar=floor(x+y);
col=sum(ar);
br(1)=ar(1);
for i=2:m
br(i)=br(i-1)+ar(i);
end
cols=randperm(nrxy);
p1(1:m)=0;
for j=1:m
if (ar(j)==0) continue; end
if (j==1) temp=0;
else temp=br(j-1);
end
for i=1:nrx
if ((cols(i)>temp) & (cols(i)<=br(j))) p1(j)=p1(j)+1; end
end; end
p2=ar'-p1;
switch sel
case 1
dum=euclideandis(p1,p2);
case 2
dum=manhattandis(p1,p2);
case 3
dum=chebyshovdis(p1,p2);
case 4
dum=correcoeffdiff(p1,p2);
end
if (abs(dum)>=abs(dxy))
fr=fr+1;
end
end
pvalue=fr/sim;
```

```
function distance = euclideandis(x,y)      %x and y: two vectors to be tested.
if (max(size(x))~=max(size(y)))
error('Array sizes do not match.');
end
if ((min(size(x))~=1) | (min(size(y))~=1))
error('Both x and y are vectors');
end
distance =sqrt(sum((x-y).^2))/max(size(x));

function distance = manhattandis(x,y)      %x and y: two vectors to be tested.
if (max(size(x))~=max(size(y)))
error('Array sizes do not match.');
end
if ((min(size(x))~=1) | (min(size(y))~=1))
error('Both x and y are vectors');
end
distance =sum(abs(x-y))/max(size(x));

function distance = chebyshovdis(x,y)      %x and y: two vectors to be tested.
if (max(size(x))~=max(size(y)))
error('Array sizes do not match.');
end
if ((min(size(x))~=1) | (min(size(y))~=1))
error('Both x and y are vectors');
end
distance =max(abs(x-y));

function diff = correcoeffdiff(x,y)      %x and y: two vectors to be tested.
m=max(size(x));
if (m~=max(size(y)))
error('Array sizes do not match.');
end
if ((min(size(x))~=1) | (min(size(y))~=1))
error('Both x and y are vectors');
end
xbar=mean(x);
ybar=mean(y);
aa=sum(x.*y)-ybar*sum(x)-xbar*sum(y)+m*xbar*ybar;
bb=sum(x.^2)-2*xbar*sum(x)+m*xbar^2;
cc=sum(y.^2)-2*ybar*sum(y)+m*ybar^2;
diff =1-aa/sqrt(bb*cc);
```

2.1.3 Statistic test of between-network difference

Known the weight pairs (a_{ij}, b_{ij}), i, j=1, 2, ...,m, i.e., (A_1, B_1). Assume the difference A_1-B_1 between two vectors A_1 and B_1 follows the normal distribution. If there is not statistic difference between network matrices A and B, the mean of A_1-B_1 should be zero. The t-test can be used to test the statistic difference (Zhang, 2018). The Matlab codes are as follows

```
AA=input('Input the excel file name of network matrix A: ','s');
A=xlsread(AA);
BB=input('Input the excel file name of network matrix B: ','s');
B=xlsread(BB);
m=size(A,1); mm=size(A,2); n=size(B,1); nn=size(B,2);
if ~((m==mm) & (n==nn) & (m==n))
error('Network matrices A and B should be the square matrices of the same size.');
end
% Matrices A and B are the square matrices of the same size.
alpha=input('Input significance level(e.g., 0.01)');
A1=reshape(A, m*m, 1);    %Transform matrix A to vector A1
B1=reshape(B, m*m, 1);    %Transform matrix B to vector B1
[h,p,ci,stats]=ttest(A1-B1,0,alpha,0);
if (h==1) fprintf(['Difference is statistically significant (p=' num2str(p) ')\n']);
else fprintf(['Difference is not statistically significant (p=' num2str(p) ')\n']); end
```

2.2 Boolean weights based network matrix

If network matrices A and B are Boolean type, i.e., the weights a_{ij} and b_{ij} take 0 or 1, point correlation can be used to calculate between-network similarity/correlation (Zhang, 2015, 2017, 2018). In this method, network matrices A and B are firstly transformed to vectors A_1 and B_1 respectively, i.e., (A_1, B_1)={(a_{ij}, b_{ij}) | i, j=1, 2, ..., m}, and point correlation is thus calculated and tested.

Generalized Matlab codes for Point correlation between two network matrices, A and B, are as follows (Zhang, 2015, 2017, 2018)

```
AA=input('Input the excel file name of network matrix A: ','s');
A=xlsread(AA);
BB=input('Input the excel file name of network matrix B: ','s');
B=xlsread(BB);
m=size(A,1); mm=size(A,2); n=size(B,1); nn=size(B,2);
if ~((m==mm) & (n==nn) & (m==n))
error('Network matrices A and B should be the square matrices of the same size.');
end
% Matrices A and B are the square matrices of the same size.
sig=input('Input the significance level (e.g., 0.05): ');
x=reshape(A, m*m, 1);    %Transform matrix A to vector x
y=reshape(B, m*m, 1);    %Transform matrix B to vector y
n=size(x,1);
aa=sum((x==0) & (y==0));
bb=sum((x==0) & (y~=0));
```

```
cc=sum((x~=0) & (y==0));
dd=sum((x~=0) & (y~=0));
pointcorr=(aa*dd-bb*cc)/sqrt((aa+bb)*(cc+dd)*(aa+cc)*(bb+dd));
chi2=n*(aa*dd-bb*cc)^2/((aa+bb)*(cc+dd)*(aa+cc)*(bb+dd));
chi2test=chi2>chi2inv(1-sig,1);
%chi2=10.8 for sig=0.001; chi2=6.635 for sig=0.01; chi2=7.87 forsig=0.005
fprintf(['Point correlation=' num2str(pointcorr) '\n']);
if (chi2test==1) fprintf(['Point correlation is statistically significant (p=' num2str(sig) ')\n']); end
if (chi2test==0) fprintf(['Point correlation is not statistically significant (p=' num2str(sig) ')\n']); end
```

2.3 Other generalized methods

2.3.1 Degree change index

Degree change index, proposed by Zhang (2012b), can be popularized to compare two networks

$$D=\sum_{j=1}^{m}(|O_{Aj}-O_{Bj}|+|I_{Aj}-I_{Bj}|)$$

where D: value of degree change index; m: total number of nodes in the network; O_{Aj}, O_{Bj}: out-degree of node for networks A and B respectively; I_{Aj}, I_{Bj}: in-degree of node j for networks A and B respectively.

2.3.2 Weight change index

Weight change index (S), popularized from parameter robustness (Zhang, 2016), can be used to compare two networks

$$S=\sum_{i}\sum_{j}|a_{ij}-b_{ij}|$$

2.3.3 Eigenvector matrix change index

Suppose the eigenvector matrices of the network matrices A and B are V_a and V_b respectively. The eigenvector matrix change index is defined as

$$V=\sum_{i}\sum_{j}|V_{aij}-V_{bij}|$$

where V_{aij} and V_{bij} are elements of the eigenvector matrices V_a and V_b respectively, i, j=1, 2, ..., m.

```
AA=input('Input the excel file name of network matrix A: ','s');
A=xlsread(AA);
BB=input('Input the excel file name of network matrix B: ','s');
B=xlsread(BB);
m=size(A,1); mm=size(A,2); n=size(B,1); nn=size(B,2);
if ~((m==mm) & (n==nn) & (m==n))
error('Network matrices A and B should be the square matrices of the same size.');
end
% Matrices A and B are the square matrices of the same size.
S=sum(sum(abs(A-B)));
[VA,D] = eig(A,'nobalance');
[VB,D] = eig(B,'nobalance');
```

```
V=sum(sum(abs(VA-VB)));
fprintf([' Weight change index S=' num2str(S) '\n']);
fprintf(['Eigenvector matrix change index V=' num2str(V) '\n']);
```

3 Discussion

In practical uses, several methods above can be jointly used to analyze various differences or similarities between two networks. It should be noted that all methods above can be improved, in particular the degree change index, weight change index, and eigenvector matrix change index. For instance, these absolute indices can be standardized as some forms of proportions.

Finally, I assume that the two network matrices are known, which should be defined and specified by researchers.

Acknowledgment

We are thankful to the support of The National Key Research and Development Program of China (2017YFD0201204), and Discovery and Crucial Node Analysis of Important Biological and Social Networks (2015.6-2020.6), from Yangling Institute of Modern Agricultural Standardization.

References

Dormann CF. 2011. How to be a specialist? Quantifying specialisation in pollination networks. Network Biology, 1(1): 1-20

Manly BFJ. 1997. Randomization, Bootstrap and Monte Carlo Methods in Biology (2nd Edition). Chapman & Hall, London, UK

Schoenly KG, Zhang WJ. 1999. IRRI Biodiversity Software Series. V. RARE, SPPDISS, and SPPANK: programs for detecting between-sample difference in community structure. IRRI Technical Bulletin No. 5. International Rice Research Institute, Manila, Philippines

Solow AR. 1993. A simple test for change in community structure. Journal of Animal Ecology, 62: 191-193

Zhang WJ. 2011a. A Java algorithm for non-parametric statistic comparison of network structure. Network Biology, 1(2): 130-133

Zhang WJ. 2011b. A Java program for non-parametric statistic comparison of community structure. Computational Ecology and Software, 1(3): 183-185

Zhang WJ. 2012a. Computational Ecology: Graphs, Networks and Agent-based Modeling. World Scientific, Singapore

Zhang WJ. 2012b. Several mathematical methods for identifying crucial nodes in networks. Network Biology, 2(4): 121-126

Zhang WJ. 2012c. How to construct the statistic network? An association network of herbaceous plants constructed from field sampling. Network Biology, 2(2): 57-68

Zhang WJ. 2015. Calculation and statistic test of partial correlation of general correlation measures. Selforganizology, 2(4): 65-77

Zhang WJ. 2016. Network robustness: Implication, formulization and exploitation. Network Biology, 6(4): 75-85

Zhang WJ. 2017. Network pharmacology of medicinal attributes and functions of Chinese herbal medicines: (II) Relational networks and pharmacological mechanisms of medicinal attributes and functions of Chinese herbal medicines. Network Pharmacology, 2(2): 38-66

Zhang WJ. 2018. Fundamentals of Network Biology. World Scientific Europe, London, UK

Zhang WJ, Li X. 2015a. General correlation and partial correlation analysis in finding interactions: with Spearman rank correlation and proportion correlation as correlation measures. Network Biology, 5(4): 163-168

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

Zhang WJ, Schoenly KG. 1999. IRRI Biodiversity Software Series III. BOUNDARY: A program for detecting boundaries in ecological landscapes. IRRI Technical Bulletin No. 3.International Rice Research Institute, Manila, Philippines

Zhang WJ, Zhan CY. 2011. An algorithm for calculation of degree distribution and detection of network type: with application in food webs. Network Biology, 1(3-4): 159-170