Article

Detecting connectedness of network: A Matlab program and application in tumor pathways and a phylogenic 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 9 December 2015; *Accepted 17 January 2016*; *Published 1 December 2016*

Abstract
Two nodes in a network are called to be connected if there is a path between the two nodes. In a connected network, each pair of nodes is connected. In this article, I present full Matlab codes of the vertex-fusion algorithm to detect network connectedness. In the program, the number of connected components and all nodes of each connected component will be displayed. It is expected for using in some studies of network pharmacology.

Keywords network; connectedness; vertex-fusion algorithm; Matlab.

Selforganizology
ISSN 2410-0080
URL: http://www.iaees.org/publications/journals/selforganizology/online-version.asp
RSS: http://www.iaees.org/publications/journals/selforganizology/rss.xml
E-mail: selforganizology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Two nodes (vertices) *u* and *v* in a network (graph) *X* are called to be connected, if there is a path between *u* and *v* (Zhang, 2012). For a given network *X*, the node set

$$V=\bigcup_{i=1}^{m} V_i$$

Two node *u* and *v* in V_i is connected, if and only if u, $v \in V_i$. The induced subnetworks $X[V_i]$, i=1,2,...,m, are defined as the connected components of *X*. That is, in the connected network *X*, each pair of nodes is connected (Zhang, 2012). Otherwise *X* is an unconnected network. *X* is a connected network if and only if for each of the classification that divides node set *V* into two nonempty subsets V_1 and V_2, there always exists a link, and one of its node is in V_1 and another node is in V_2. If *X* is an unconnected network, then each of its connected components is the maximal connected subnetwork of *X*. In present article I give the full Matlab codes of the vertex-fusion algorithm to detect network connectedness. In this program, the number of connected components and all nodes of each connected component are displayed.

2 Algorithm

The vertex-fusion algorithm for detecting network (graph) connectedness is that for the adjacency matrix of a network, starting from a node (vertex), fusing all its adjacent nodes, and then fusing the new added nodes adjacent to it, until no new node adjacent to it is added. A connected component is thus obtained. When fusing the nodes v_i and v_j, add row *j* of adjacency matrix to row *i*, and add column *j* to column *i*, and then delete row *j* and column *j*. In this way, all connected components are obtained (Chan et al., 1982; Zhang, 2012).

The following are Matlab codes of the function, connDet.m, for calculating connectedness of a network. The program will display the number of connected components and all nodes of each connected component (Zhang, 2012)

```
%Zhang WJ. 2016. Detecting connectedness of network: A Matlab program and application in tumor pathways.
  Selforganizology, 3(4):
function conn=connDet(d)
%Calculate connectivity of graph. g[]: if node i belongs to j-th connected component then g[i]=j.
%d[][]: adjacency matrix, d=(dij)v*v, where v 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; i, j=1,2,…, v.
%conn: string containing number of connected components and all nodes of each connected component.
v=size(d,1);
g=zeros(1,v+1);
for i=1:v
g(i)=0;
end
s=1;
t=1;
while (v>0)
g(t)=s;
a1=0;
for j=1:v
a1=a1+d(j,t);
end
m=t+1;
while (v>0)
for h=m:v
if ((d(t,h)==0) | (g(h)~=0)) continue; end
g(h)=s;
for i=m:v
if (d(i,h)==0) continue; end
d(i,t)=1;
d(t,i)=1;
end; end
a2=0;
for j=1:v
```

```
a2=a2+d(j,t);
end
if ((a1-a2)<0)
a1=a2;
continue;
end
if ((a1-a2)>=0) break; end
end
lab=0;
for k=m:v
if (g(k)==0) break; end
if (k==v) lab=1; break; end
end
if (lab==1) break; end
t=k;
s=s+1;
end
conn='';
conn=strcat(conn,'Number of connectivity components in the network: ',num2str(s),'\n');
conn=strcat(conn,'Node----Belonged connectivity component\n');
for i=1:v
conn=strcat(conn,num2str(i),'----', num2str(g(i)),'\n');
end
conn=strcat(conn,'\n');
```

3 Application

Use the data of adjacency matrices of tumor pathways, p53, RAS, VEGF, PPAR (Zhang, 2016). The results of network connectedness detection show that there is only one connected component in these tumor pathways.

Use the data of adjacency matrices of world's 54 human races and populations (nodes) (Zhang and Qi, 2014; Zhang, 2015). The nodes No. 20 and 21 belong to a component, and the remaining 52 nodes belong to another component.

Acknowledgment

We are thankful to the support of 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, Discovery and Crucial Node Analysis of Important Biological and Social Networks (2015.6-2020.6), from Yangling Institute of Modern Agricultural Standardization, and Project on Undergraduate Teaching Reform (2015.7-2017.7), from Sun Yat-sen University, China.

References

Chan SB, et al. 1982. Graph Theory and Its Applications. Science Press, Beijing, China

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. A node degree dependent random perturbation method for prediction of missing links in the network. Network Biology, 6(1): 1-11

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