Article

Finding the shortest tree in the network: A Matlab program and application in tumor pathway

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 October 2016; *Accepted 20 November 2016*; *Published online 1 March 2017*

Abstract
In present article, I present full Matlab codes of Kruskal algorithm for calculating the shortest tree and use it in tumor pathway.

Keywords network; shortest tree; algorithm; Matlab.

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
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

In a network, the sum of weights of all tree branches of a tree is defined as weight of the tree. A tree with minimal weight is defined as the shortest tree, i.e., minimum spanning tree. Occasionally we need to find the shortest tree in a network. In present article, I will present full Matlab codes of Kruskal algorithm for calculating the shortest tree and use it in tumor pathways.

2 Algorithm

Suppose a network *X* has *v* nodes and *e* links, and between-node weights matrix of the network is $a=(a_{ij})$, $i, j=1,2,\ldots,v$, where a_{ij} is the weight between the nodes *i* and *j*. A method to find the shortest tree is (Lu and Lu, 1995; Zhang, 2012):

(1) Use a spanning tree arbitrarily.

(2) Add a link of a cotree to form a circuit. In the circuit if there is an link which is longer than the link added, then replace the longer link with the new added link and thus achieve a new tree. Repeat this process until no more the longer links.

To find the shortest tree, we always use Kruskal algorithm. In Kruskal algorithm, first check the links of *X* from smaller weight link to larger weight link, add these links to *T* based on the principle of not generating any loop, until the number of links of *T* is equal to the number of links of *X*-1. The procedures are as follows (Chan et al., 1982; Zhang, 2012):

(1) Order the links in the edge set, from smaller weight link to larger weight link, as e_1, e_2,⋯, e_v.

(2) Let T={e_1}, i=1,and j=2.

(3) If i=v-1, print T, and terminate calculation, otherwise, return (4).

(4) If a circuit is generated after e_i is added to T, let j=j+1, return (4), otherwise return (5).

(5) Let T=T∪{e_i}, j=j+1, i=i+1, return (3).

The following are Matlab codes, Kruskal.m, for Kruskal algorithm:

```
%Kruskal algorithm to calculate the shortest tree in a network.
a=input('Input the file name of between-node weights matrix of the weighted network (e.g., adj.xls, etc. The matrix is a=(aij)v*v, where v is the number of nodes in the network. aij is the weight between the nodes i and j, i, j=1,2,…, v): ','s');
a=xlsread(a);
v=size(a,1);
cc=0;
t=zeros(v); t1=zeros(v); b=zeros(1,100000);
k=1;
for i=1:v-1
for j=i+1:v
if (a(i,j)>0)
b(k)=a(i,j);
kk=1;
for l=1:k-1
if (b(k)==b(l))
kk=0;
break;
end; end
k=k+kk;
end; end; end
k=k-1;
for i=1:k-1
for j=i+1:k
if (b(j)<b(i))
cc=b(j);
b(j)=b(i);
b(i)=cc;
end; end; end
m=0;
for l=1:k
if (m==v) break; end
for i=1:v-1
for j=i+1:v
if (a(i,j)==b(l))
t(i,j)=b(l);
t(j,i)=b(l);
```

```
for i1=1:v;
for j1=1:v;
t1(i1,j1)=t(i1,j1);
end; end
while(v>0)
in=1;
c=0;
for i1=1:v
kk=0;
for j1=1:v
if (t1(i1,j1)>0)
kk=kk+1;
c=j1;
end; end
if (kk==1)
t1(i1,c)=0;
t1(c,i1)=0;
in=0;
end
end
if (in~=0) break; end
end
in=0;
for i1=1:v-1
for j1=i1+1:v
if (t1(i1,j1)>0)
in=1;
break;
end; end; end
if (in~=0)
t(i,j)=0;
t(j,i)=0;
else m=m+1; end
end; end; end; end
fprintf(['Shortest tree:' '\n']);
tree=t
```

The software can be found in supplementary material.

3 Application Example

Use Kruskal algorithm and the adjacency matrix of tumor pathway p53 (Huang and Zhang, 2012; Li and Zhang, 2013), the calculated shortest tree in the p53 pathway is

(2,5), (3,5), (4,5), (5,6), (5,7), (2,8), (7,9), (2,10), (2,12), (2,14), (16,18), (4,28), (16,29), (29,31), (35,37), (32,40), (38,41), (38,42), (40,42), (40,43), (24,47), (26,47), (32,47), (33,47), (34,47), (35,47), (36,47), (39,47), (44,47), (45,47), (46,47), (16,48), (21,49), (23,49), (25,49), (27,49), (48,49), (18,50), (20,50), (22,50), (24,50), (49,51), (1,52), (4,52), (11,52), (13,52), (15,52), (17,52), (19,52), (30,52), (48,52)

References

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

Huang JQ, Zhang WJ. 2012. Analysis on degree distribution of tumor signaling networks. Network Biology, 2(3): 95-109

Li JR, Zhang WJ. 2013. Identification of crucial metabolites/reactions in tumor signaling networks. Network Biology, 3(4): 121-132

Lu KC, Lu HM. 1995. Graph Theory and Its Applications (2nd edition). Tsinghua University Press, Beijing, China

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