Article

Matlab algorithm to generate adjacency matrix from connection pairs that nodes are represented by strings

WenJun Zhang[1], **Yanhong Qi**[2]
[1]School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
[2]Libraries of Sun Yat-sen University, Sun Yat-sen University, Guangzhou, China
E-mail: zhwj@mail.sysu.edu.cn, qiyh@mail.sysu.edu.cn

Received 12 November 2019; *Accepted 18 December 2019*; *Published 1 December 2020*

Abstract
It is harder to generate an adjacency matrix from connection (i.e., link) pairs that nodes are represented by strings than by numerical ID numbers. In present article, we developed a Matlab algorithm to generate adjacency matrix from connection pairs in which nodes are represented by strings. Full codes and executable program of the algorithm were given.

Keywords Matlab algorithm; connection pairs; connection weights; adjacency matrix; transformation.

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

Adjacency matrices are widely used in many algorithms of graph theory and network science (Zhang, 2015, 2016a-k, 2017a-d, 2018; Zhang and Li, 2016; Zhang and Feng, 2017). On the other hand, we have usually collected connection pairs, in which nodes are always represented by strings (names, abbreviations, ABC codes, etc.). Nevertheless, it is much harder to generate an adjacency matrix from connection (i.e., link) pairs that nodes are represented by strings than by numerical ID numbers. In this article, I present a Matlab algorithm to generate adjacency matrix from connection pairs in which nodes are represented by strings. Full codes and executable program of the algorithm are given also.

2 Method

Suppose there are *num* connection pairs, which are recorded in a space separated text file as, for example

$$\begin{matrix} v_1 & v_9 \\ v_1 & v_{20} \\ v_3 & v_5 \\ \cdots \end{matrix} \quad (1)$$

or

v_1 v_5 w_1
v_1 v_2 w_2
v_2 v_9 w_3
…

(2)

where v_i are nodes represented by strings and w_i are connection weights (numerical values), i=1,2,…. We want to find the series of unique nodes in (1) or (2):

$$v_1, v_2, \ldots, v_m \qquad (3)$$

where m is the total number of unique nodes, and transform (1) to an adjacency matrix $d=(d_{ij})_{m\times m}$. d_{ij}=1, if two nodes v_i and v_j are adjacent (connected), and d_{ij}=0, if v_i and v_j are not adjacent; i, j=1,2,…, m; or transform (2) to an adjacency matrix $d=(d_{ij})_{m\times m}$. $d_{ij}=w_{ij}$, if two nodes v_i and v_j are adjacent (connected), and d_{ij}=0, if v_i and v_j are not adjacent; i, j=1,2,…, m. Or for the situation (2), if a threshold for the connection weight, h>0, is given, we have d_{ij}=1 if $w_{ij}\geq h$ and, v_i and v_j are adjacent (connected), and d_{ij}=0 if $w_{ij}<h$ or v_i and v_j are not adjacent; i, j=1,2,…, m.

The IDs of rows and columns of the resultant adjacency matrix correspond to the natural IDs of unique nodes in the resultant series of unique nodes.

According to the above principle, the full Matlab algorithm, used in Matlab environment, is developed as the following.

```
[newdel,OK]=listdlg('liststring',{'Node-Node Data','Node-Node-Weight Data'},'listsize',[300
100],'OkString','OK','CancelString','Cancel','promptstring','Two Columns or Three Columns Data?','selectionmode','single');
if (newdel==2)
[choice,OK]=listdlg('liststring',{'Generate Weight Matrix','Generate 0-1 Adjacency Matrix'},'listsize',[300
100],'OkString','OK','CancelString','Cancel','promptstring','Two Columns or Three Columns Data?','selectionmode','single');
if (choice==2)
threshold=input('Input the threshold of the weight to produce 0 or 1 (=1 if >=weight, and =0 if <weight): ');
end
end
if (newdel==2)
strr=input('Input the text file name of node-node-weight matrix w (w=(wij)num×3): ','s');
[w(:,1) w(:,2) w(:,3)]=textread(strr,'%s%s%s');
elseif (newdel==1)
strr=input('Input the text file name of node-node matrix w (w=(wij)num×2): ','s');
[w(:,1) w(:,2)]=textread(strr,'%s%s');
end
num=length(w(:,1));
[st,m]=uniqNodes(w);
mm=0;
for i=1:num
for k=1:m
if (isequal(w{i,1},st{k})==1)
```

```
wid(i,1)=k;
end
if (isequal(w{i,2},st{k})==1)
wid(i,2)=k;
end
end
if (newdel==2)
wid(i,3)=str2num(w{i,3});
end
end
d=zeros(m,m);
for k=1:num
if (newdel==1) d(wid(k,1),wid(k,2))=1; d(wid(k,2),wid(k,1))=1;
elseif ((choice==1) && (newdel==2)) d(wid(k,1),wid(k,2))=wid(k,3); d(wid(k,2),wid(k,1))=wid(k,3);
elseif ((choice==2) && (newdel==2)) d(wid(k,1),wid(k,2))=wid(k,3)>threshold; d(wid(k,2),wid(k,1))=wid(k,3)>threshold;
end
end
fprintf('Matrix of Interactions\n');
d
fprintf(['Names of nodes from ID 1 to ' num2str(m) ':\n']);
st

function [st,m]=uniqNodes(w)
num=length(w(:,1));
w2=[w(:,1);w(:,2)];
m=1;
st{1}=w2{1};
for i=2:2*num
s=0;
for j=1:m
if isequal(w2{i},st{j})==0
s=s+1;
end
end
if (s==m)
m=m+1;
st{m}=w2{i};
end
end
```

The executable GUI software (see supplementary material) of the algorithm above is partly indicated in Fig. 1.

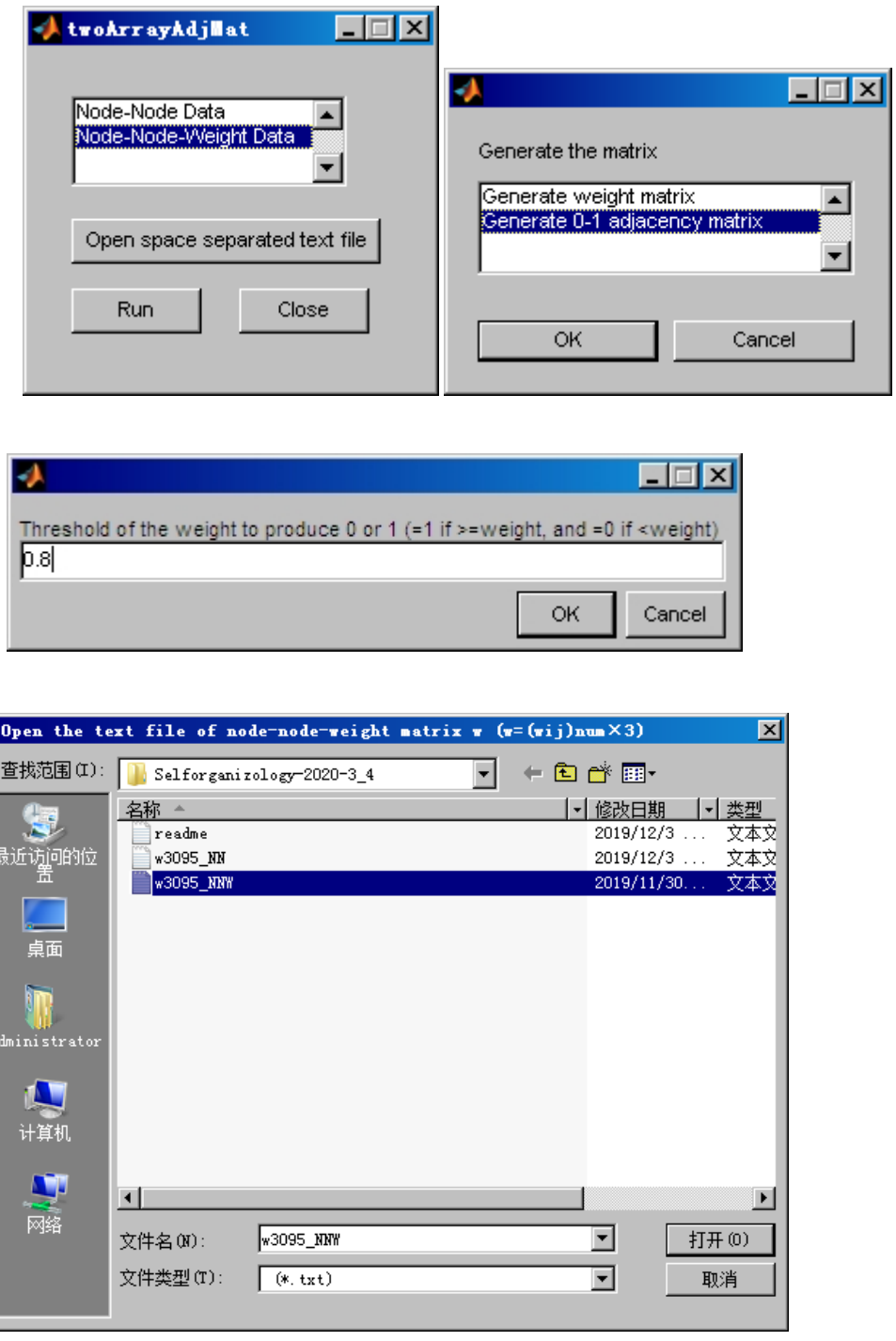


Fig. 1 The executable GUI software of the algorithm.

3 Application Example

There is a dataset for connection pairs with connection weights, as indicated in Fig. 2.

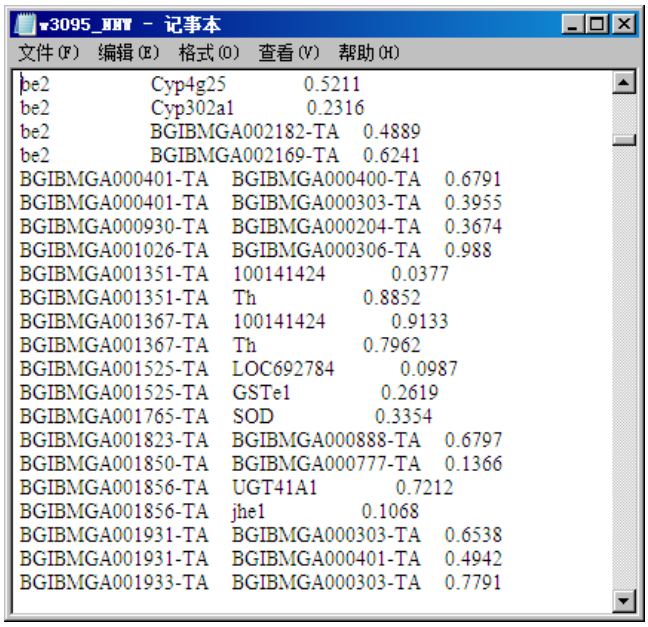
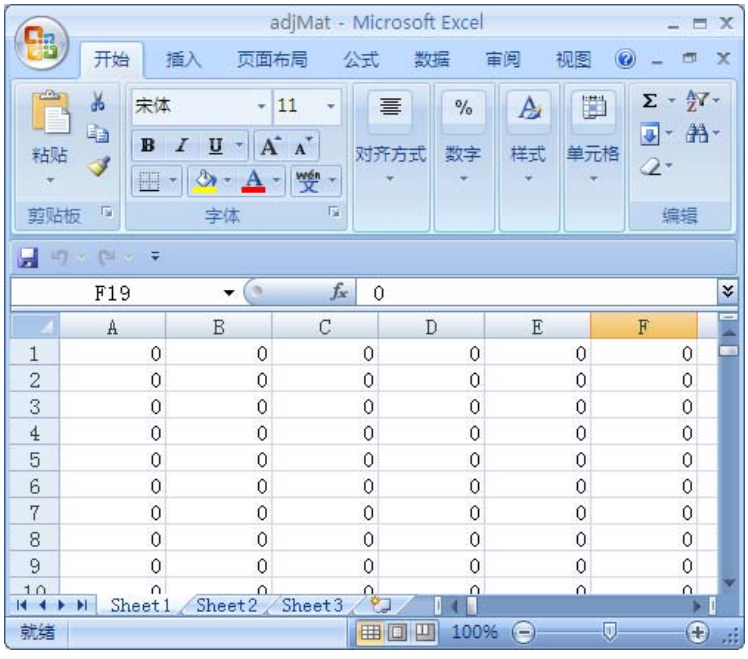
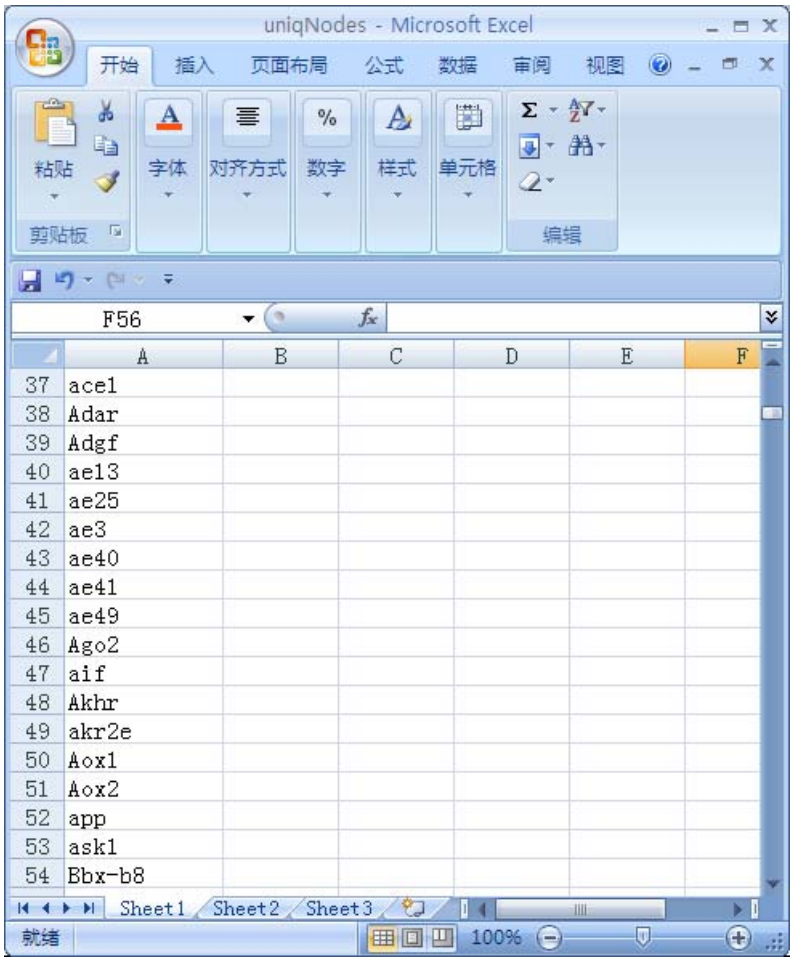


Fig. 2 An example dataset for connection pairs with connection weights. The dataset is stored in a space separated text file.

In the software interface, choose Node-Node-Weight Data and a dialog appears. In the dialog. choose generate 0-1 adjacency matrix and an input dialog appears for inputting the threshold *h*, for example, 0.8 is entered. The data file is loaded by open the space separated text file dialog. Finally, run the software (Fig. 1) and the two sets of results (adjacency matrix and unique node series) can be saved in to two specified excel files (Fig. 3a and 3b).



(3a)



(3b)

Fig. 3 Two sets of results, adjacency matrix (3a), and unique node series (3b).

4 Disscusion

It is obvious that the algorithm is applicable to datasets that nodes are represented by both strings and numerical ID numbers.

If the dataset is stored in an excel file. Users can save the file as a space separated text file (.prn), and open the ".prn" text file and re-save it as the text file (.txt) that can be used in the algorithm.

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

Zhang WJ. 2015. Prediction of missing connections in the network: A node-similarity based algorithm. Selforganizology, 2(4): 91-101

Zhang WJ. 2016a. Detecting connectedness of network: A Matlab program and application in tumor pathways and a phylogenic network. Selforganizology, 3(4): 117-120

Zhang WJ. 2016b. A mathematical model for dynamics of occurrence probability of missing links in predicted missing link list. Network Pharmacology, 1(4): 86-94

Zhang WJ. 2016c. A node-similarity based algorithm for tree generation and evolution. Network Biology, 6(3): 55-64

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

Zhang WJ. 2016e. How to find cut nodes and bridges in the network? A Matlab program and application in tumor pathways. Network Pharmacology, 1(3): 82-85

Zhang WJ. 2016f. 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): 54-65

Zhang WJ. 2016g. Finding trees in the network: Some Matlab programs and application in tumor pathways. Network Pharmacology, 1(2): 66-73

Zhang WJ. 2016h. A Matlab program for finding shortest paths in the network: Application in the tumor pathway. Network Pharmacology, 1(1): 42-53

Zhang WJ. 2016i. A node degree dependent random perturbation method for prediction of missing links in the network. Network Biology, 6(1): 1-11

Zhang WJ. 2016j. A random network based, node attraction facilitated network evolution method. Selforganizology, 3(1): 1-9

Zhang WJ. 2016k. 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: with application in finding sub-networks in tumor pathways. Proceedings of the International Academy of Ecology and Environmental Sciences, 6(1): 13-23

Zhang WJ. 2017a. Maximum matching of the network: A Matlab program and application. Selforganizology, 4(4): 65-68

Zhang WJ. 2017b. Finding minimum cost flow in the network: A Matlab program and application. Selforganizology, 4(2): 30-34

Zhang WJ. 2017c. Finding fundamental circuits in the network: A Matlab program and application in tumor pathway. Selforganizology, 4(1): 14-17

Zhang WJ. 2017d. Finding the shortest tree in the network: A Matlab program and application in tumor pathway. Network Pharmacology, 2(1): 13-16

Zhang WJ, Feng YT. 2017. Metabolic pathway of non-alcoholic fatty liver disease: Network properties and robustness. Network Pharmacology, 2(1): 1-12

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