Article

Generate networks with power-law and exponential-law distributed degrees: with applications in link prediction of tumor pathways

WenJun Zhang[1], **Xin Li**[2]
[1]School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
[2]College of Plant Protection, Northwest A & F University, Yangling 712100, China; Yangling Institute of Modern Agricultural Standardization, Yangling 712100, China
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org, lixin57@hotmail.com

Received 13 August 2015; *Accepted 20 September 2015*; *Published online 1 March 2016*

Abstract
In present study I proposed a method for generating biological networks based on power-law ($p(x)=x^{-\lambda}$) and exponential-law ($p(x)=e^{-\lambda x}$) distribution functions. Given the parameter of power-law or exponential-law distribution function, λ, the algorithm generates an expected frequency distribution according to the given parameter, thereafter creates an adjacency matrix in which (practical) frequency distribution of node degrees matches the expected frequency distribution. The results showed that power-law distribution function performs much better than exponential-law distribution function in generating networks. Using the revised algorithm, tumor related networks (pathways) are simulated and predicted. The results prove that the algorithm is overall effective in predicting network links (14.6%~21.2% of correctly predicted links against 0.1%~3.4% of that for random assignments). Matlab codes of the algorithms are given also.

Keywords power-law; exponential-law; degree distribution; adjacency matrix; network generation; link prediction.

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 1999, Barabasi and Albert proposed a well-known mechanism for network evolution. Cancho and Sole (2001) presented an algorithm, which can generate a variety of complex networks with diverse degree distributions. Zhang (2011, 2012a, 2012b, 2012c, 2013, 2015a, 2016a, 2016b), Zhang and Liu (2012) proposed a series of methods and models for network generation and evolution. In present study, I will propose a method for generating biological networks based on power-law and exponential-law distribution functions. Power-law and exponential-law distribution, in particular power-law distribution, is the most popular form of degree distribution of various networks. Therefore, the present method is of significant in the network analysis. Link prediction aims to estimate the likelihood of the existence of a connection between two nodes based on observed connections and the attributes of nodes (Zhang, 2015d; Zhou, 2015). Many biological networks, such as food webs, protein–protein interaction networks and metabolic networks, are incomplete due to missing

links. For example, 80% of the molecular interactions in cells of Yeast (Yu et al., 2008) and 99.7% of human (Amaral, 2008) are still unknown. An incomplete network occurs due to our limited knowledge on a complete network, or the network is in evolution and thus more connections or even nodes are expected with time. Link (connection) prediction can considerably reduce the experimental costs for connection finding, and the algorithms can be used to predict the connections that may appear in the future of evolving networks (Lü and Zhou, 2011; Lü et al., 2012; Zhou, 2015). So far, numerous papers on this topic have been published (Clauset et al., 2008; Guimera and Sales-Pardo, 2009; Barzel and Barabási, 2013; Bastiaens et al., 2015; Lü et al., 2015; Zhang, 2015b, 2015c, 2015d, 2016a, 2016b; Zhang and Li, 2015; Zhao et al., 2015; Zhou, 2015). In present study, I will use the proposed algorithm to approach its effectiveness in link prediction of biological networks.

2 Algorithms

2.1 Generation of the network with power-law or exponential-law distributed degrees

The power-law and exponential-law distribution functions are as follows (Goemann, 2011; Zhang, 2011; Zhang, 2012a; Fig.1)

$$p(x) = x^{-\lambda}$$

$$p(x) = e^{-\lambda x}$$

where λ is a known constant. Suppose there are v nodes in the network being built. First, set the permitted error of degree distribution, er (e.g., 0.05), and the maximum number of simulations, sim (e.g., v*5), where the fitting error of degree distribution is defined as

$$error = \sum_{i=1}^{m} | f_i - ff_i | / \sum_{i=1}^{m} f_i$$

where f_i and ff_i are the expected and practical frequencies of nodes in the degree interval $[in_i, in_{i+1})$ (i=1,2,…, m-1).

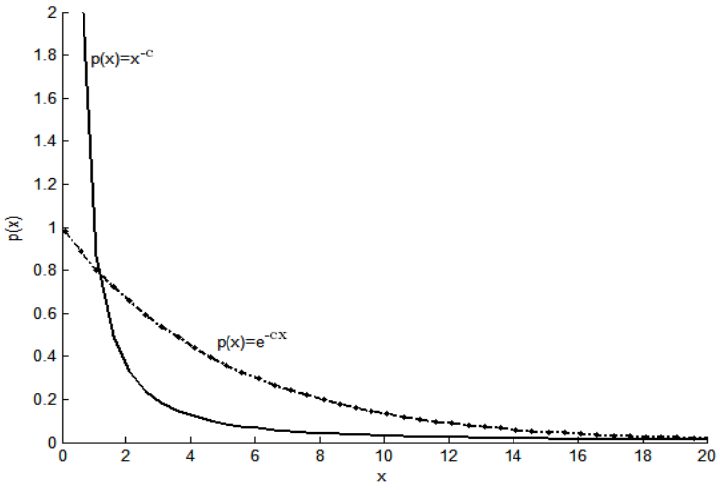

Fig. 1 Illustration of power-law and exponential-law distribution functions.

The algorithm generates an expected frequency distribution according to the given parameter of power-law or exponential-law distribution function, and then creates an adjacency matrix in which (practical) frequency distribution of node degrees coincides with the expected frequency distribution. The adjacency matrix of the network being generated is d=(d_{ij}), i, j=1,2,…,v, where d_{ij}=d_{ji}, d_{ii}=0, and if d_{ij}=1 or d_{ji}=1, there is

a connection between nodes i and j. There are $v(v\text{-}1)/2$ unknown variables d_{ij}, i, 1,2,…,v-1; $j>i$. However, there are only v known conditions (degrees of v nodes), s_i ,i=1,2,…, v. Therefore there are in general multiple solutions for $d=(d_{ij})$. I thus use the enumerating/meriting method to find the $d=(d_{ij})$ that meets the error requirement.

The procedures of the algorithm are as the following

(1) Let fa=1.2, and $fac=fa$.

(2) Find degree intervals [in_i, in_{i+1}), i=1,2,…, m-1, where

in_1=1;
$in_{i+1}=in_i*fac$, (for power-law distribution)
$in_{i+1}=in_i*fac*2$, (for exponential-law distribution)
i=1, 2,…, m-1
$in_m=v$, if $in_m>v$

(3) Let $class=m$-1. Calculate the probability of degree distribution, i.e., the probability of a node's degree falling in the interval [in_i, in_{i+1})

$$p_i = in_i^{-\lambda} - in_{i+1}^{-\lambda} \quad \text{(for power-law distribution)}$$
$$p_i = e^{-\lambda in_i} - e^{-\lambda in_{i+1}} \quad \text{(for exponential-law distribution)}$$

and the corresponding frequencies $f_i=p_i*v$, i=1,2,…, $class$-1. Calculate

$$su = \sum_{i=1}^{class-1} f_i$$

and let $f_{class}=v\text{-}su$.

(4) Let initial error, $miner=10^{10}$, and simulation times, tot=1.

(5) Let

$$s = \sum_{i=1}^{class} f_i$$

(6) Calculate node degrees

$$s_k = floor(\sum_{i=1}^{class}\sum_{j=1}^{f_i}(in_{i+1} - in_i) * rand + in_i)$$

where $floor(x)$ denotes the integer part of x, $rand$ is a random value in (0,1), k=1,2,….

(7) Rearrange v pairs of (node, degree), from greater to smaller in terms of node degrees

v_i, s_i
v_j, s_j
…
v_q, s_q

In this step, for each node v_i with s_i expected connections, randomly create v_i–m_i connections to other nodes and each of v_i–m_i nodes has a connection, where m_i is the number of connections already created by

previous nodes.

(8) Finally, produce the candidate adjacency matrix, $d=(d_{ij})$, calculate the practical frequency distribution, ff_i, i=1, 2,…, and the *error*.

(9) If there is at least one node degree, e.g., s_r=0, return (6). Otherwise, go to (10).

(10) If the calculated *error*<*miner*, let *minerr*=*error*, *dd*=*d*, $ffd_i=ff_i$, go to (11), or else go to (12).

(11) If *minerr*≤*er*, go to (14).

(12) If *tot*≥*sim*, go to (13), or else let *tot*=*tot*+1, return (5).

(13) Let *fac*=*fac*+0.3, return (2). If *fac*>*v*, go to (14).

(14) Print adjacency matrix (*dd*), expected and practical frequency distributions (f_i and ffd_i, i=1,2,…, m), and node degrees.

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

```
%Reference: Zhang WJ. 2016. Generate networks with power-law and exponential-law distributed degrees: with applications in link prediction of tumor pathways. Network Pharmacology, 1(1): 15-35
v=input('Input the number of nodes in the network: ');
typedis=input('Input the the type of frequency distribution of degrees (1: Power-law distri (F(x)=1-x^(-c)); 2: Exponential-law distri (F(x)=1-e^(-cx))): ');
c=input('Input the parametrical value (c) of frequency distribution of node degrees: ');
disp('e.g., c=1.5 for power-law distri (mostly falls in (0, 2]), and c=0.2 for exponential-law distri. For power-law distri, the following formula can be used to obtain a suitable c, c=1.6347-0.1401m+0.0019v+0.0038m^2, r^2=0.83, p=0.0006<0.01, where m is the mean of node degrees, v is the number of nodes in the network.')
er=0.05;
sim=v*5;
fa=1.2;    % It can be replaced with, e.g., fa=22.6712-34.4086*c+0.0764*v-0.0423*c*v+13.0632*c^2
for fac=fa:0.3:v
in=zeros(1,v);
if (typedis==1)
i=1;
in(1)=1;
while (v>0)
i=i+1;
in(i)=in(i-1)*fac;
if (in(i)>v) in(i)=v; break; end
end; end
if (typedis==2)
i=1;
in(1)=1;
while (v>0)
i=i+1;
in(i)=in(i-1)*fac*2;
if (in(i)>v) in(i)=v; break; end
end; end
deg=in(1:i)
class=i-1;
```

```
ff=zeros(1,class);
p=zeros(1,class);
su=0;
for i=1:class-1
p(i)=netConstrDistr(typedis,c,in(i),in(i+1));
ff(i)=p(i)*v;
su=su+ff(i);
end
ff(class)=v-su;
id=0;
minerr=1e+10;
tot=1;
while (v>0)
[adj0,fff0,error0]=netGen(class,deg,ff);
if (error0<minerr)
adj=adj0; fff=fff0; minerr=error0;
if (minerr<=er) id=1; break; end
end
if (tot>=sim) break; end;
tot=tot+1;
end
if (id==1) break; end
end
fprintf('\nAdjacency matrix of the generated network\n')
disp([adj])
fprintf('\n')
fprintf('\nIntervals of node degrees\n')
for i=1:class
fprintf(['[' num2str(deg(i)) ',' num2str(deg(i+1)) ')    ])
end
fprintf('\n\nExpected frequency distribution of node degrees of the generated network\n')
disp([ff])
fprintf('\n')
fprintf('Practical frequency distribution of node degrees of the generated network\n')
disp([fff])
fprintf(['Fitting error of expected and practical distribution of node degrees of the generated network: ' num2str(minerr) '\n'])
fprintf(['\nNode degrees of the generated network\n' num2str(sum(adj)) '\n'])
fprintf(['\nMean of node degrees of the generated network: ' num2str(mean(sum(adj))) '\n\n\n'])
```

The functions, netConstrDistr.m, and netGen.m, are as follows

```
function pab=netConstrDistr(typedis,lamda,a,b)
pab=0;
if (typedis==1) pab=a^(-lamda)-b^(-lamda); end
if (typedis==2) pab=exp(-lamda*a)-exp(-lamda*b); end
```

```
function [adj,pracff,error]=netGen(m,in,expff)
%pracff[]: degree distri of produced network; er: expected chi-square of practical and expected degree distri
%in[]: degree threshold pairs vector; expff[]: frequency; sim: simulation times
%[in[i],in(i+1)): expff(i), i=1, 2, ..., m; m is the vector dimension
s=sum(expff);
while (m>0)
adj=zeros(s);
pracff=zeros(1,m);
f=zeros(1,s);
p=zeros(1,s);
w=zeros(1,s);
k=0; u=0;
for i=1:m
for j=1:expff(i)
k=k+1;
f(k)=floor((in(i+1)-in(i))*rand()+in(i));
u=u+f(k);
end; end
for i=1:s
p(i)=i;
end
for i=1:s-1
k=i;
for j=i:s-1
if (f(j+1)>f(k)) k=j+1; end
end
l=p(i); p(i)=p(k); p(k)=l;
u=f(i); f(i)=f(k); f(k)=u;
end
for i=1:s
lab=0;
vv=0;
for j=1:i-1
if (adj(j,i)==1)
adj(i,j)=1;
vv=vv+1;
if (vv==f(i)) lab=1; break; end
end; end
if (lab==1) continue; end
cc=0;
for j=1:s
w(j)=j;
end
while (f(i)~=0)
```

```
cs=floor((s-cc)*rand()+1);
if (w(cs)==i) continue; end
if ((adj(w(cs),i)==0) & (w(cs)<i)) continue; end
adj(i,w(cs))=1;
if (cs<s-cc)
for j=cs+1:s-cc
w(j-1)=w(j);
end; end
cc=cc+1;
if (cc>=(f(i)-vv)) break; end
end; end
for i=1:s
for j=1:s
if (adj(j,i)~=adj(i,j)) adj(j,i)=1; adj(i,j)=1; end
end; end
for i=1:s
p(i)=0;
for j=1:s
if (adj(i,j)==1) p(i)=p(i)+1; end
end; end
for i=1:m
pracff(i)=0;
for j=1:s
if ((p(j)>=in(i)) & (p(j)<in(i+1))) pracff(i)=pracff(i)+1; end
end; end
error=sum(abs(expff-pracff))/sum(expff);
isonodes=sum(sum(adj)==0);
if (isonodes==0) break; end
end
```

2.2 Link prediction of the network

Given the adjacency matrix of an original network, i.e., the network with which missing links are prepared for predicting. Suppose the mean of node degrees of the original network is *m'*. The procedures of the algorithm are as follows

(1) Let the mean of node degrees of generated network *D*, $m=m'(1+per)$, where *per* is the perturbation rate, and *per*=0.2, 0.3, etc., which represents a percentage increment of mean in the network perturbation or evolution.

(2) Use *m* to empirically calculate parameter λ (see the section 3.2, eq. (1)).

(3) Run the algorithm above (section 2.1, with power-law distribution), the resultant adjacency matrix, *G*, is then achieved.

(4) Swap rows and columns of *G* according to the ranking of node degrees of *D*, such that the same row (column) of *G* and *D* has the same ranking of node degree in the respective matrices. By doing so, the final adjacency matrix of the predicted network, *H*, transformed from *G*, is achieved.

(5) Compare *H* and *D*, the connection pairs in original network only, connection pairs in predicted (i.e.,

generated) network only, and connection pairs in both original and predicted networks, etc., are thus achieved.

(6) If simulation times are achieved, go to (7); or else return (3).

(7) Calculate means of all indices and mean number (likelihood) of links in original network only, mean number (likelihood) of links in generated network only, and mean number (likelihood) of links in both networks. The percentage of correctly predicted links *vs*. true links is defined as

$$\frac{x}{x+y}\times 100$$

where x is the number of links in both networks, and y is the number of links in original network only. The percentage of correctly predicted links *vs*. predicted links is defined as

$$\frac{x}{x+z}\times 100$$

where x is the number of links in both networks, and z is the number of links in predicted (generated) network only.

The following are Matlab codes of the algorithm (connectionFinding.m). The functions, netConstrDistr.m, and netGen.m, are the same with the above.

```
%Reference: Zhang WJ. 2016. Generate networks with power-law and exponential-law distributed degrees: with applications in link prediction of biological networks. Selforganizology, 6(3):
clear
choice=input('Input the type (1 or 2) of data file of the network from which missing links are ready to be predicted (1: adjacency matrix; 2: two array): ');
disp('Adjacency matrix: d=(dij)m*m, where m 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,…, m');
disp('Two array: there are two columns, A1 and A2, in the data file; an element of A1 stores a node of a link and the corresponding element of A2 stores another node of the link. ');
if (choice==1)
adjstr=input('Input the file name of adjacency matrix from which missing links are ready to be predicted (e.g., raw.txt, raw.xls, etc. Adjacency matrix is d=(dij)m*m, where m 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,…, m: ','s');
end
if (choice==2)
adjstr=input('Input the file name of two array of the network from which missing links are ready to be predicted (e.g., raw.txt, raw.xls, etc. There are two columns, A1 and A2, in the data file; an element of A1 stores a node of a link and the corresponding element of A2 stores another node of the link: ','s');
end
pro=input('Input perturbation rate to increase the mean of node degrees of the network (e.g, 0.2, 0.3, etc.): ');
simu=input('Input the simulation times (e.g, 20, 30, etc.): ');
if (choice==1) adjmat=load(adjstr); v=size(adjmat,2); end
if (choice==2)
twoarray=load(adjstr);
nn=size(twoarray,1);
v=max(max(twoarray));
for i=1:nn
adjmat(twoarray(i,1),twoarray(i,2))=1;
```

```
adjmat(twoarray(i,2),twoarray(i,1))=1;
end; end
degr=sum(adjmat);
meanmat=sum(degr)/v;
m=meanmat*(1+pro);
c=1.6347-0.1401*m+0.0019*v+0.0038*m^2;
er=0.05;
sim=v*5;
fa=1.2;        %It can be replaced with, e.g., fa=22.6712-34.4086*c+0.0764*v-0.0423*c*v+13.0632*c^2-0.6
percentCorrect=zeros(1,simu);
percentPrediCorrect=zeros(1,simu);
degd=zeros(simu,v);
summ=v*(v-1)/2;
su1=zeros(summ,2*simu);
su2=zeros(summ,2*simu);
su3=zeros(summ,2*simu);
adj=zeros(v);
adjj=zeros(v);
pdegr=zeros(1,v);
pdeg=zeros(1,v);
fdegr=zeros(1,v);
fdeg=zeros(1,v);
fd=zeros(1,v);
pd=zeros(1,v);
persum=zeros(1,3);
for siml=1:simu
for fac=fa:0.3:v
in=zeros(1,v);
i=1;
in(1)=1;
while (v>0)
i=i+1;
in(i)=in(i-1)*fac;
if (in(i)>v) in(i)=v; break; end
end;
deg=in(1:i);
class=i-1;
ff=zeros(1,class);
p=zeros(1,class);
su=0;
for i=1:class-1
p(i)=netConstrDistr(1,c,in(i),in(i+1));
ff(i)=p(i)*v;
su=su+ff(i);
end
```

```
ff(class)=v-su;
id=0;
minerr=1e+10;
tot=1;
while (v>0)
[adj0,fff0,error0]=netGen(class,deg,ff);
if (error0<minerr)
adj=adj0; fff=fff0; minerr=error0;
if (minerr<=er) id=1; break; end
end
if (tot>=sim) break; end;
tot=tot+1;
end
if (id==1) break; end
end
degg=sum(adj);
for h=1:2
if (h==1) fd=degr; end
if (h==2) fd=degg; end
for i=1:v
pd(i)=i;
end
for i=1:v-1
k=i;
for j=i:v-1
if (fd(j+1)>fd(k)) k=j+1; end
end
l=pd(i); pd(i)=pd(k); pd(k)=l;
u=fd(i); fd(i)=fd(k); fd(k)=u;
end
if (h==1) pdegr=pd; fdegr=fd; end
if (h==2) pdeg=pd; fdeg=fd; end
end
for i=1:v
adjj(pdegr(i),:)=adj(pdeg(i),:);
end
for i=1:v
adj(:,pdegr(i))=adjj(:,pdeg(i));
end
degg=sum(adj);
degd(siml,:)=degg;
fprintf('\nAdjacency matrix of the original network\n')
disp([adjmat])
fprintf('\nNode degrees of adjacency matrix of the original network\n')
disp([degr])
```

```
fprintf(['\nMean of node degrees of the original network: ' num2str(mean(degr)) '\n'])
fprintf('\n\nAdjacency matrix of the generated network with power-law distributed node degrees\n')
disp([adj])
fprintf('\nNode degrees of adjacency matrix of the generated network\n')
disp([degg])
fprintf(['\nMean of node degrees of the generated network: ' num2str(mean(degg)) '\n'])
fprintf(['Fitting error of expected and practical distribution of node degrees of the generated network: ' num2str(minerr) '\n\n'])
xx=adjmat & (~adj);
yy=(~adjmat) & adj;
zz=adjmat & adj;
for i=1:3
switch i
    case 1
        mat=xx; s='Connection pairs in original network only (x)';
    case 2
        mat=yy; s='Connection pairs in predicted network only (y)';
    case 3
        mat=zz; s='Connection pairs in both original and predicted networks (z)';
end;
[pairx,pairy]=find(mat);
temp1=pairx; temp2=pairy;
pairxs=pairx(temp1<temp2); pairys=pairy(temp1<temp2);
ConnectionPairs=[pairxs pairys];
persum(i)=size(ConnectionPairs,1);
dm=size(ConnectionPairs,1);
if (i==1) su1(:,siml*2-1)=[pairxs;zeros(summ-dm,1)]; su1(:,siml*2)=[pairys;zeros(summ-dm,1)]; end
if (i==2) su2(:,siml*2-1)=[pairxs;zeros(summ-dm,1)]; su2(:,siml*2)=[pairys;zeros(summ-dm,1)]; end
if (i==3) su3(:,siml*2-1)=[pairxs;zeros(summ-dm,1)]; su3(:,siml*2)=[pairys;zeros(summ-dm,1)]; end
end
percentageCorrect(siml)=persum(3)/(persum(1)+persum(3))*100;
percentagePrediCorrect(siml)=persum(3)/(persum(2)+persum(3))*100;
disp(s)
disp([ConnectionPairs])
disp('Percentages of correctly predicted connections (%): ')
percentageCorrect(siml)
end
disp('---------------------------------Summary---------------------------------')
disp('Node degrees of original networks: ')
degr=degr
disp('Averaged node degrees of generated networks under different simulations: ')
dega=round(sum(degd)/simu+0.5)
disp('Chi square of node degrees between node degrees of original networks and averaged node degrees of generated networks: ')
chi2=sum(((degr-dega).^2)./degr)
disp('Mean percentage of correctly predicted connections vs. true connections (%): ')
meanPercent=mean(percentageCorrect)
```

```
disp('Stadard deviation of mean percentage of correctly predicted connections vs. true connections (%): ')
standardDevi=std(percentageCorrect)
disp('Mean percentage of correctly predicted connections vs. predicted connections (%): ')
meanPercentPredi=mean(percentagePrediCorrect)
disp('Stadard deviation of mean percentage of correctly predicted connections vs. predicted connections (%): ')
standardDeviPredi=std(percentagePrediCorrect)
prop1=zeros(v);
prop2=zeros(v);
prop3=zeros(v);
%su2(:,k*2-1),su2(:,k*2)
for i=1:v-1
for j=i+1:v
for k=1:simu
for l=1:v*(v-1)/2
if ((su1(l,k*2-1)==i) & (su1(l,k*2)==j)) prop1(i,j)=prop1(i,j)+1; break; end
if ((su2(l,k*2-1)==i) & (su2(l,k*2)==j)) prop2(i,j)=prop2(i,j)+1; break; end
if ((su3(l,k*2-1)==i) & (su3(l,k*2)==j)) prop3(i,j)=prop3(i,j)+1; break; end
end; end; end; end
disp('------------------------------------------------------------------------')
disp('Mean number (Likelihood) of links in both networks: ')
disp('Node        Node        Likelihood')
[pairx,pairy]=find(prop3);
s=0;
for i=1:v-1
for j=i+1:v
if (prop3(i,j)~=0) s=s+1;pairvalue(s)=prop3(i,j)/simu; end;
end; end
result=[pairx pairy pairvalue'];
ires=sortrows(result,-3);
disp([ires])
clear pairvalue
disp('Mean number (Likelihood) of links in original network only: ')
disp('Node        Node        Likelihood')
[pairx,pairy]=find(prop1);
s=0;
for i=1:v-1
for j=i+1:v
if (prop1(i,j)~=0) s=s+1;pairvalue(s)=prop1(i,j)/simu; end;
end; end
result=[pairx pairy pairvalue'];
ires=sortrows(result,-3);
disp([ires])
clear pairvalue
disp('Mean number (Likelihood) of predicted links in generated network only: ')
disp('Node        Node        Likelihood')
```

```
[pairx,pairy]=find(prop2);
s=0;
for i=1:v-1
for j=i+1:v
if (prop2(i,j)~=0) s=s+1;pairvalue(s)=prop2(i,j)/simu; end;
end; end
result=[pairx pairy pairvalue'];
ires=sortrows(result,-3);
disp([ires])
```

In link prediction, I use the data of tumor related networks (pathways) (Huang and Zhang, 2012; Li and Zhang, 2013; Pathway Central, 2012). The simulation times are set to be 20. The perturbation rate *per*=0. For comparison, the link predition with random networks (%) is conducted also.

3 Applications

3.1 Generate networks of power-law/exponential-law distributed node degrees

With different numbers of nodes, v, and distribution parameter, λ, the networks generated by the algorithm are different also (Table 1 and 2). For power-law distribution, the number of degree intervals increases as the parameter λ. The curve of exponential-law distribution function is more flat, thus the number of degree intervals seldom varies (Fig. 1, Table 2).

About the distribution parameter, λ, Goemann et al. (2011) provided a set of values. For power-law distribution, it is 1.63 (transcription network, v=279), 1.71 (signaling network, v=1571), and 1.58 (metabolic network, v=1793), respectively; for exponential-law distribution, it is 0.19 (transcription network), 0.21 (signaling network), and 0.15 (metabolic network), respectively. Zhang (2011) found that for arthropod family networks (v=66~75), λ values of power-law distribution are 1.64, 1.43, and 1.11, respectively. Based on the investigation on CSM food webs, Zhang and Zhan (2011) found that λ value of exponential-law distribution is between 0.13 and 0.30. Some ecologists found that the mean of node degrees of food webs is around 2.0. Zhang (2011) revealed that the means are 2.16, 4.07, 3.14, 2.84 and 2.19 for the networks of arthropod species respectively. Huang and Zhang (2012) found that the mean is between 2.1 and 2.9 for tumor pathways ($v\approx$20~100). Goemann et al. (2011) found that the means are 2.35 (transcription network), 2.18 (signaling network), and 3.09 (metabolic network), respectively. On average the number of connections (links) per species in a food web is roughly 2 (Cohen et al., 1990; Martinez, 1992). However, Shams and Khansari (2014) found that the means are between 4 and 15. Rahman et al. (2013) revealed that the means are between 4.68 and 10.58 for normal and cancer pathways (v=192~631). Overall power-law λ is mostly around 1.5 (1.5±0.4), and exponential-law λ is around 0.2 (0.2±0.07); the mean of node degrees is mostly around 2.6, and some exceed 4.

In general, λ values and means of node degrees for power-law distribution (Table 1) coincide with the reported findings described above.

Table 1 Generate networks with power-law distributions of node degrees.

v	**20**			**50**			**100**		
λ	**0.7**	**1.2**	**1.7**	**0.7**	**1.2**	**1.7**	**0.7**	**1.2**	**1.7**
Intervals	[1,3.6) [3.6,12.96) [12.96,20)	[1,3) [3,9) [9,20)	[1,3) [3,9) [9,20)	[1,6.3) [6.3,39.69) [39.69,50)	[1,2.4) [2.4,5.76) [5.76,13.824) [13.824,33.1776) [33.1776,50)	[1,2.1) [2.1,4.41) [4.41,9.261) [9.261,19.4481) [19.4481,40.841) [40.841,50)	[1,10.2) [10.2,100)	[1,2.1) [2.1,4.41) [4.41,9.261) [9.261,19.4481) [19.4481,40.841) [40.841,85.7661) [85.7661,100)	[1,1.5) [1.5,2.25) [2.25,3.375) [3.375,5.0625) [5.0625,7.5938) [7.5938,11.3906) [11.3906,17.0859) [17.0859,25.6289) [25.6289,38.4434) [38.4434,57.665) [57.665,86.4976) [86.4976,100)
Expected freq. distri.	11.8414 4.8305 3.3282	14.6484 3.9196 1.4320	16.9102 2.6124 0.4773	36.2142 9.9848 3.8010	32.5130 11.3711 3.9769 1.3909 0.7481	35.8356 10.1518 2.8759 0.8147 0.2308 0.0912	80.3220 19.6780	58.9478 24.1994 9.9344 4.0783 1.6742 0.6873 0.4787	49.8068 24.9996 12.5481 6.2983 3.1613 1.5868 0.7964 0.3998 0.2007 0.1007 0.0506 0.0509
Practical freq. distri.	12 5 3	15 4 1	17 3 0	36 11 3	33 12 4 1 0	36 11 3 0 0	81 19	61 24 10 4 1 0 0	51 25 13 7 3 1 0 0 0 0 0 0
Fitting error	0.033	0.043	0.048	0.041	0.046	0.045	0.014	0.042	0.047
Node degrees	17 15 13 8 7 4 3 3 3 5 2 4 3 1 3 3 3 2 3 2	17 6 5 2 2 2 3 2 2 2 1 2 1 3 1 2 1 1 2 1	8 6 2 2 2 2 2 1 1 2 1 1 1 2 1 2 1 1 3 1	46 45 43 25 16 15 13 12 11 9 9 8 5 5 5 5 5 5 5 6 5 5 5 5 4 5 4 5 4 5 5 6 6 4 6 4 4 4 4 4 4 4 5 4 4 8 6 3 7 6	16 12 9 8 5 5 5 4 3 3 3 3 2 2 2 2 2 2 2 2 2 3 1 1 1 2 1 2 2 2 1 2 1 3 2 1 2 1 1 1 1 6 2 3 1 3	8 6 4 4 4 3 3 3 3 2 2 2 2 2 2 1 3 1 1 2 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 4 1 2 2 2 1 1 1 2 1 3 1 1 5 3	81 76 73 65 64 52 51 45 43 36 31 33 26 22 17 13 12 11 9 10 9 9 9 10 9 9 9 10 9 9 8 8 9 7 9 8 8 7 7 7 10 11 10 7 7 6 9 7 9 6 5 8 10 8 9 9 7 8 9 7 8 6 8 5 7 5 6 7 10 6 7 8 6 6 4 7 7 8 8 10 7 8 9 8 7 4 9 10 7 6 5 5 4 8 8 10 8 10 6 8	24 17 17 16 12 9 8 7 7 6 6 5 5 4 4 4 3 3 3 3 3 3 3 3 3 3 2 3 2 4 3 2 4 2 2 2 2 5 2 2 1 1 1 2 1 1 2 2 2 2 1 1 4 3 2 1 1 1 2 1 1 2 1 1 2 2 2 2 4 2 1 1 3 1 1 1 1 1 3 1 2 1 1 2 1 3 1 1 2 2 1 2 2 2 2 2	8 6 6 5 4 4 4 4 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 1 2 1 1 1 3 1 1 1 1 3 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1 4 1 1 3 1 1 3 1 1 1 1 1 2 1 1 4 1 1 1 1 7
Mean of node degrees	5.2	2.9	2.1	8.8	2.9	2.1	13.9	3.2	1.9

Table 1 (continue) Generate networks with power-law distributions of node degrees.

v	**200**			**300**		
λ	**0.7**	**1.2**	**1.7**	**0.7**	**1.2**	**1.7**
Intervals	[1,15.3) [15.3,200)	[1,5.1) [5.1,26.01) [26.01,132.651) [132.651,200)	[1,1.5) [1.5,2.25) [2.25,3.375) [3.375,5.0625) [5.0625,7.5938) [7.5938,11.3906) [11.3906,17.0859) [17.0859,25.6289) [25.6289,38.4434) [38.4434,57.665) [57.665,86.4976) [86.4976,129.7463)	[1,18.6) [18.6,300)	[1,5.1) [5.1,26.01) [26.01,132.651) [132.651,300)	[1,1.5) [1.5,2.25) [2.25,3.375) [3.375,5.0625) [5.0625,7.5938) [7.5938,11.3906) [11.3906,17.0859) [17.0859,25.6289) [25.6289,38.4434) [38.4434,57.665) [57.665,86.4976) [86.4976,129.7463)

			[129.7463,194.6195) [194.6195,200)			[129.7463,194.6195) [194.6195,291.9293) [291.9293,300)
Expected freq. distri.	170.3690 29.6310	171.6896 24.3030 3.4401 0.5673	99.6136 49.9993 25.0962 12.5966 6.3226 3.1735 1.5929 0.7995 0.4013 0.2014 0.1011 0.0507 0.0255 0.0257	261.2330 38.7670	257.5344 36.4545 5.1602 0.8509	149.4204 74.9989 37.6443 18.8949 9.4840 4.7603 2.3893 1.1993 0.6020 0.3021 0.1517 0.0761 0.0382 0.0192 0.0193
Practical freq. distri.	166 34	168 29 3 0	99 51 27 13 6 3 1 0 0 0 0 0 0 0	255 45	252 43 5 0	147 78 40 20 8 5 1 1 0 0 0 0 0 0 0
Fitting error	0.044	0.047	0.033	0.042	0.044	0.045
Node degrees	184 178 140 113 110 105 104 99 97 92 88 85 83 79 77 78 62 58 41 40 38 37 33 32 26 21 18 17 17 14 14 14 12 14 14 14 13 14 13 13 14 14 14 14 14 14 14 14 14 13 13 13 16 13 13 13 13 11 12 12 12 12 12 12 11 11 11 11 14 11 12 11 11 13 14 11 10 10 12 11 10 13 15 10 12 10 10 10 10 12 11 9 13 12 13 11 13 10 13 12 11 12 13 12 13 13 12 10 12 9 15 13 13 10 12 13 11 12 12 8 11 11 18 8 13 14 13 11 13 10 12 12 10 12 11 12 11 8 9 11 10 13 13 7 11 7 18 11 9 9 13 9 15 9 12 12 8 13 10 13 10 9 11 15 12 8 13 15 13 9 14 15 12 8 11 10 9 8 15 10 9 7 9 11 12 15 10 9 16 12 14 10 9 11 9 17 9 9 9 12	55 35 27 25 24 23 21 20 20 20 19 19 16 16 15 15 13 12 12 11 11 10 8 8 7 6 5 5 5 5 4 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 4 4 5 5 5 3 3 5 4 3 3 3 3 3 8 2 3 3 3 3 3 3 4 3 3 5 5 5 3 4 4 4 4 4 5 4 3 4 3 2 5 4 5 4 5 3 3 3 5 2 4 3 2 3 2 2 4 2 2 3 3 2 3 2 4 5 2 2 5 2 2 4 2 2 3 3 2 2 4 4 5 2 3 3 3 2 1 4 2 3 2 2 1 6 3 3 1 2 3 6 2 4 3 4 3 3 2 2 3 1 2 3 3 2 4 3 2 2 1 4 4 6 4 8 3 4 5 2	14 10 9 9 7 6 6 6 6 6 5 5 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 1 1 1 1 1 3 1 1 2 2 1 4 1 1 1 1 1 2 1 1 1 1 1 1 2 4 3 1 1 1 1 2 1 1 2 1 1 3 1 1 1 1 1 1 2 1 1 3 1 1 1 1 1 2 2 1 2 1 1 1 1 1 1 3 1 1 1 1 1 1 5 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 4 1 1 2 1 2 2 1 2 1 3 2 1 2 1 1 1 1 2 1 1 1 1 2 2 1 3 2 1 1 4 1 1 1 1 3 1 1 3 1 1 1 1 1 1 4 2 2 3 4 2 1	294 261 256 219 220 209 209 204 185 180 173 161 156 132 123 117 111 103 88 86 87 84 79 78 62 52 51 53 50 43 38 36 31 29 29 27 26 26 18 16 18 17 18 17 18 17 18 18 18 17 17 17 17 16 17 17 17 17 17 17 15 16 16 17 16 16 16 16 20 16 16 16 17 16 16 16 15 18 15 15 15 15 15 15 15 15 15 15 14 17 15 15 15 15 16 14 14 14 14 16 19 14 14 14 17 18 14 13 16 16 17 13 13 13 14 17 17 15 13 13 13 13 16 15 15 18 15 12 16 14 12 16 15 12 14 16 16 12 16 12 15 12 16 15 13 15 13 11 18 17 16 17 13 15 12 14 15 15 14 14 14 10 14 13 11 16 13 13 15 13 17 14 18 14 14 16 15 20 16 18 13 17 13 18 15 14 13 18 18 13 14 18 15 16 14 14 14 16 15 13 13 15 10 14 18 13 14 15 11 17 17 15 12 18 12 16 18 15 19 14 16 12 18 16 10 16 14 14 12 14 15 12 15 12 16 15 25 16 12 12 20 14 18 14 14 17 14 16 15 15 13 13 15 12 14 15 21 14 13 13 14 14 14 12 17 16 14 13 8 11	107 91 76 48 38 25 24 24 23 21 20 19 16 16 17 17 17 15 14 14 14 14 12 13 13 11 11 9 9 8 7 8 8 7 7 5 6 6 5 5 5 5 5 5 5 6 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 5 4 5 4 4 4 4 4 6 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 4 4 4 4 4 5 5 6 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 4 3 5 3 5 4 3 3 5 4 3 5 3 5 3 3 4 3 3 3 4 3 3 3 3 3 4 3 3 3 3 5 3 3 4 5 3 3 3 3 4 6 3 3 5 5 4 3 3 3 3 3 3 3 2 3 4 5 4 4 2 4 5 4 5 5 4 3 3 4 3 2 3 2 4 2 4 2 3 4 4 2 2 3 2 2 2 5 3 5 3 2 3 4 3 2 2 2 2 4 3 3 3 2 2 2 4 4 2 6 6 2 2 2 5 2 5 2 5 1 4 5 5 2 1 3 4 3 2 1 2 3 5 6 1 2 4 4 5 5 2 3 4 1 6 2 2 5 2 5	24 14 11 10 8 8 8 7 7 7 6 6 6 5 5 5 5 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 3 3 1 1 1 4 1 2 2 2 1 2 1 1 3 1 1 1 3 1 1 1 1 1 2 1 1 3 1 1 2 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 1 3 1 2 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 2 4 1 3 1 1 1 1 3 1 4 1 1 3 3 3 1 1 1 1 1 1 1 1 1 2 1 2 1 5 1 4 3 1 1 1 2 2 2 2 1 2 1 4 1 1 1 1 2 1 2 2 1 2 1 1 1 1

													12	16	18	17	4	3	2	3	1	3	1	3
													12	18	13	15	1	3	2	3	1	3	1	6
													12	17	13	17	3	2	2	2	3	1	3	1
													14	14	18	12	2	2	3	3	3	1	1	1
													16	14	8	14	2	5	3	1	3	2	4	2
													13	18	15	14	6	8	1	2	1	6	3	1
													18	9	15	14	11	3	1	2				
													14	18			5	1	3	4				
																	4							
Mean of node degrees	20.8				5.4				2.1				27.7				5.8				2.1			

Table 1 (continue) Generate networks with power-law distributions of node degrees.

v	**500**																								
λ	**1.2**													**1.7**											
Intervals	[1,7.2)	[7.2,51.84)	[51.84,373.248)	[373.248,500)										[1,2.1) [2.1,4.41) [4.41,9.261) [9.261,19.4481) [19.4481,40.841) [40.841,85.7661) [85.7661,180.1089) [180.1089,378.2286) [378.2286,500)											
Expected freq. distri.	453.2081	42.4130	3.9692	0.4098										358.3563 0.6538 0.0207	101.5178	28.7587	8.1470	2.3079	0.1852	0.0525					
Practical freq. distri.	443	54	3	0										350 0	107	34	7	2	0	0	0				
Fitting error	0.046													0.043											
Node degrees	218	132	105	50	50	47	45	44	43	43	42	42		35	20	16	15	15	15	15	13	12	9	9	9	9
	42	40	40	38	36	36	35	35	34	34	33	30		8	8	8	7	7	7	7	7	6	6	6	6	6
	30	29	28	25	21	21	21	18	18	17	17	14		6	6	6	5	5	5	5	5	5	4	4	4	4
	11	9	9	8	8	8	7	7	7	7	7	7		4	4	4	4	4	4	4	4	4	4	4	4	4
	7	7	7	7	7	7	7	7	7	7	7	6	5	3	3	3	3	3	3	3	3	3	3	3	3	3
	6	7	6	6	6	6	6	6	6	7	6	6	6	3	3	3	3	3	3	3	3	3	3	3	3	4
	6	6	6	6	6	6	6	6	6	6	6	6	6	3	3	3	3	3	3	3	3	3	3	3	3	3
	6	6	6	7	6	6	6	6	6	6	9	6	6	2	3	2	2	2	2	2	2	2	2	2	2	2
	6	6	6	6	6	6	6	6	6	6	6	6	6	2	3	2	2	2	2	2	2	2	2	2	2	2
	6	6	6	6	6	6	6	6	6	6	7	6	6	2	3	2	2	2	2	2	2	2	2	2	2	2
	6	6	6	6	5	5	5	9	5	5	6	5	5	2	2	2	2	2	2	2	2	2	3	2	2	2
	5	5	5	5	5	7	5	8	5	5	5	5	7	2	2	2	2	2	2	2	4	2	2	2	2	2
	5	6	5	5	5	7	5	5	5	5	5	5	5	2	2	2	3	2	2	2	2	2	2	2	2	3
	5	5	5	7	5	5	6	5	5	5	5	5	5	2	2	2	2	1	1	3	1	1	2	2	2	1
	5	5	5	5	5	5	5	5	5	5	5	5	5	1	1	3	5	2	1	3	1	1	1	1	3	1
	6	6	5	6	6	5	5	5	5	4	5	5	5	1	1	1	2	1	1	1	1	1	2	1	1	1
	5	7	5	5	5	6	5	5	5	5	7	6	5	4	1	1	1	1	1	4	1	1	1	2	3	2
	9	4	4	6	10	4	4	4	4	4	4	4	6	1	1	2	1	1	1	1	1	1	2	2	1	1
	5	4	5	5	4	5	4	4	4	6	4	8	5	1	2	1	1	2	1	1	3	1	1	1	1	3
	9	5	5	4	6	4	5	5	6	4	8	4	4	4	1	3	1	2	2	1	1	1	3	1	1	1
	5	4	4	5	5	6	4	4	4	7	8	4	6	2	4	1	3	1	1	1	1	1	1	1	3	2
	4	5	4	4	4	4	7	4	5	4	5	6	4	1	2	1	1	5	2	2	2	1	3	2	1	2
	4	4	9	4	4	6	5	7	4	4	3	6	4	1	1	1	1	2	4	1	2	1	3	1	1	1
	4	3	3	3	6	2	5	5	5	9	4	3	3	3	1	1	4	2	2	3	1	1	1	1	1	1
	3	7	6	5	6	3	10	4	3	3	3	4	4	1	1	2	1	2	1	1	2	1	1	1	1	2
	5	6	5	3	6	3	3	6	6	3	5	4	5	2	3	3	2	2	2	1	1	1	1	1	2	1
	3	3	3	3	3	3	3	6	3	3	5	6	3	2	1	2	2	1	1	1	1	1	1	1	1	2
	3	7	6	7	3	7	6	2	7	4	7	5	6	1	1	1	2	1	1	2	3	1	1	1	3	2
	3	4	6	3	7	3	6	2	2	3	7	6	6	1	1	3	1	1	1	3	3	1	1	1	2	1
	3	2	3	8	6	6	2	6	6	4	5	3	6	1	1	4	1	1	2	1	1	2	1	1	1	1
	4	5	5	3	3	5	6	5	3	6	3	2	2	1	1	1	3	2	3	1	1	1	1	1	1	5
	3	4	5	4	3	3	5	6	5	3	5	2	3	1	2	4	1	1	1	5	2	2	1	1	1	1
	7	5	5	7	3	5	5	6	3	5	3	2	3	1	1	1	1	1	3	1	3	1	1	2	1	1
	3	7	5	6	4	5	4	7	4	3	6	5	3	5	1	2	1	1	2	1	1	1	1	1	1	1
	7	5	2	6	9	3	5	4	2	7	5	4	4	3	1	2	4	3	3	1	2	1	2	1	2	1
	6	2	5	4	3	4	5	4	4	4	5	6	3	2	1	1	1	3	3	1	1	1	1	1	1	4
	3	3	5	4	6	7	3	4	4	5	3	2	2	3	1	1	1	1	5	1	1	3	2	2	1	4
	4	3	1	4	3	5	6	3	4	5	8	5	1	1	2	1	5	1	1	1	1	1	1	2	2	1
	6	3	5	5	4	4	6	1	1	1				1	1	2	1	2	5							
Mean of node degrees	7.8													2.4											

The adjacency matrix of the network of power-law distribution with v=20, and λ=1.7 is

0	1	1	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	1	0
1	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 2 Generate networks with exponential-law distributions of node degrees.

v	**20**			**50**			**100**		
λ	**0.1**	**0.2**	**0.3**	**0.1**	**0.2**	**0.3**	**0.1**	**0.2**	**0.3**
Intervals	[1,5.4) [5.4,20)	[1,5.4) [5.4,20)	[1,4.8) [4.8,20)	[1,10.2) [10.2,50)	[1,9.6) [9.6,50)	[1,8.4) [8.4,50)	[1,16.2) [16.2,100)	[1,13.2) [13.2,100)	[1,13.8) [13.8,100)
Expected freq. distri.	6.4418 13.5582	9.5827 10.4173	10.0778 9.9222	27.2121 22.7879	33.6062 16.3938	33.0179 16.9821	70.6939 29.3061	74.7369 25.2631	72.4895 27.5105
Practical freq. distri.	6 14	10 10	10 10	26 24	33 17	32 18	70 30	76 24	72 28
Fitting error	0.044	0.042	0.008	0.048	0.024	0.041	0.014	0.025	0.01
Node degrees	19 15 13 12 10 11 8 8 8 7 6 8 6 4 5 5 7 3 5 4	18 14 12 10 8 9 10 9 8 6 5 4 3 5 3 4 5 5 3 5	13 12 8 8 7 7 6 5 5 6 4 2 2 4 2 4 4 2 3 2	49 45 43 42 37 31 26 27 24 24 23 19 19 17 17 17 12 12 13 11 10 12 10 9 9 10 10 9 11 12 10 9 7 9 8 12 10 9 10 10 8 9 10 8 8 10 9 10 6 10	47 46 41 34 30 29 25 24 24 14 17 13 11 13 10 9 9 9 9 8 8 7 9 10 7 6 6 6 7 8 6 7 7 7 9 10 8 8 6 6 8 4 7 4 7 8 6 6 6 4	47 39 28 30 26 25 18 17 17 14 14 16 14 9 9 9 8 9 7 7 7 8 6 6 5 5 6 9 6 5 6 7 5 6 6 5 6 5 5 6 7 4 6 8 6 7 5 5 5 4	99 97 94 93 77 76 69 66 59 57 52 46 45 47 40 34 32 36 30 29 24 26 24 18 19 19 18 17 16 15 15 13 14 15 15 14 14 13 14 13 12 11 13 17 16 15 11 13 13 10 11 10 12 14 12 10 14 12 11 10 15 11 13 16 11 12 11 16 14 13 11 14 11 12 14 11 15 13 13 14 11 13 11 13 11 9 14 13 13 9 12 14 12 17 9 12 10 15 13 12	99 96 88 84 79 78 69 66 59 42 39 44 39 36 36 28 28 22 19 17 17 13 13 15 13 13 12 12 10 11 11 10 12 11 9 13 10 10 11 12 9 12 11 11 14 9 10 8 11 13 10 13 10 9 10 12 9 9 10 8 11 13 10 9 10 11 7 11 9 12 13 9 9 13 11 12 8 14 11 12 12 12 10 10 12 13 10 10 12 10 8 8 11 7 11 7 13 8 8 9	92 90 79 77 68 65 61 52 46 43 33 33 27 28 27 23 24 22 23 21 20 20 18 15 16 15 12 13 12 13 13 12 13 13 12 11 11 10 10 10 10 10 9 10 8 9 11 9 8 9 8 8 10 12 10 12 13 9 12 12 10 12 7 12 7 13 10 9 10 10 12 10 10 9 14 12 8 10 10 7 7 14 10 9 13 10 9 9 7 8 11 7 10 11 11 9 9 12 11 7
Mean of node degrees	8.2	7.3	5.3	15.8	12.6	10.8	22.6	19.3	17.9

3.2 Relationship between distribution parameter λ, mean of node degrees, number of nodes

For power-law distribution, the following empirical formulae can be used to estimate a suitable λ in using the

algorithm, or estimate the mean of node degrees, which were derived from the results of running the algorithm (Table 1)

λ=1.6347-0.1401m+0.0019v+0.0038m^2, r^2=0.83, p=0.0006<0.01 (1)

m=36.5867-57.9304λ+0.1108v-0.0578λv+22.1056λ^2, r^2=0.93, p=0<0.01 (2)

where m is the mean of node degrees, v is the number of nodes in the network.

3.3 Application in link prediction of tumor related networks (pathways)

Some of the summarized results for link prediction of tumor related network (pathways) are listed in Table 3, in which the mean percentages of correctly predicted connections with random networks (%) are given also.

Compared to the mean percentages of correctly predicted connections *vs*. true connections with random networks, it is obvious that the results of generated networks are effective (Zhou, 2015). Therefore the algorithm is effective in predicting missing links of biological networks.

Table 3 Simulation of some tumor related networks (pathways).

	Ras	p53	Akt	HGF	JAK-STAT	JNK	PPAR	TGF-β	TNF
Mean percentage of correctly predicted connections *vs*. true connections with **random networks** (%) (±standard deviation)	**3.4±2.3**	**0.1±0.3**	**3.3±1.7**	**2.6±2.4**	**2.4±2.0**	**2.6±1.6**	**3.1±2.7**	**2.7±2.5**	**0.4±0.9**
Mean percentage of correctly predicted connections *vs*. true connections with **generated networks** (%) (±standard deviation)	**20.8±7.2**	**16.8±4.5**	**17.7±3.7**	**17.4±5.4**	**14.6 ±4.0**	**21.2±3.7**	**15.9±5.4**	**15.8±5.6**	**17.8±4.9**
Node degrees of original network	1 2 2 3 10 1 1 1 2 2 2 3 2 1 2 2 2 1 2 1 2 3 4 1 1 2 2 3 1 2 2 3 2 2 3	1 5 1 3 5 1 3 2 2 2 1 2 1 2 1 3 1 2 1 1 1 1 1 2 1 2 1 2 2 1 2 2 1 1 2 1 2 2 1 3 1 2 1 1 1 1 13 5 6 6 3 17	1 1 3 5 1 1 1 1 1 2 1 7 1 2 1 1 1 1 1 2 2 1 4 1 1 1 2 1 1 2 3 1 1 2 1 1 1 1 1 1 2 1 1 8 32 1 1 2 1 1 1 1	3 1 1 2 2 2 1 1 1 2 2 3 2 2 2 2 2 3 8 2 1 2 1 2 1 2 1 1 1 6 1 2 2 3 2 2	3 5 2 9 2 2 2 3 2 2 2 2 2 3 6 3 3 1 2 2 2 2 2 1 2 2 3 1 2 7 1 4 2 3 2 2 2 2 2 2 3 4 1 4	5 1 2 1 6 3 10 3 2 2 11 2 10 2 3 3 1 3 2 2 4 3 2 3 2 5 21 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 13	1 1 2 1 2 3 2 2 2 3 2 2 4 2 1 2 2 4 2 4 2 2 2 4 3 3 2	1 6 5 1 1 1 3 1 3 1 1 1 4 1 4 1 2 2 2 3 2 2 1 2 3 3 2 1 3 1 5 2 3 2 2 1 4 1	4 2 2 1 3 1 3 4 3 5 3 6 2 4 1 6 1 2 2 2 2 2 2 2 2 1 2 2 2 2 4

Mean node degrees of generated networks																		
2	3	2	6	2	2	5	2	4	6	5	2	2	2	2	15	5	3
3	7	2	4	6	7	2	3	4	15	3	2	2	2	8	2	3	2
21	2	6	2	3	3	3	3	2	2	5	4	3	4	2	2	3	2
2	2	4	4	3	3	2	2	2	4	8	3	2	2	4	2	3	5
3	3	3	3	2	4	2	4	3	3	3	3	3	4	4	2	3	7
3	5	2	3	3	8	3	7	2	2	9	3	3	3	2	2	3	18
3	2	2	4	3	4	3	3	2	4	6	3	18	3	6	2	3	4
3	3	2	4	2	2	3	3	7	4	4	4	2	3	4	2	2	8
3	2	2	4	4	4	3	5	4	2	2	4	3	8	3	3	2	3
3	2	2	2	4	4	17	3	3	3	3	3	3	6	3	4	3	3
3	5	2	2	4	3	2	3	3	3	4	4	2	2	3	3	3	3
8	2	2	4	6	3	2	3	3	3	3	4	2	5	2	3	2	2
2	3	2	4	3	3	2	3	3	2	3	5	3	3	4	4	2	2
3	4	2	4	4	3	2	2	2	2	20	2	2		3	2	2	2
2	3	3	2	3	4	2	9	2	2	2	2			4	2	2	2
2	4	3	3	4	3	2	3	2	8	2	2			7	3	4	
2	2	2	2	2	4	3	4	2	6	2	2			4	3		
3		3	2	2	2	2		2	4	3	3			2	2		
		3	3	2	2			2	2	3	3			4	2		
		2	4	2	2			4	4	3	3						
		2	3	2	2			2	5	3	2						
		2	2	2	2					2	2						
		8	5	4	2					2	11						
		7	7	2	8												
		4	15	15	2												
				2	4												
				2	2												
				2	2												

4 Discussion

For power-law distribution, to reduce the time cost in running the algorithm, the constant, *fa*, can be approximated with a fitting value (derived from Table 1)

$$fa'=22.6712-34.4086\lambda+0.0764v-0.0423\lambda v+13.0632\lambda^2,\ r^2=0.93,\ p=0<0.01 \quad (3)$$

Further, let $fa=fa'-0.6$. Perturbation rate is a relatively defined parameter. In the practical uses of link prediction, the perturbation rate, *per*, can be set to be 0.2, 0.5, etc. In addition, the relationship between distribution parameter λ, mean of node degrees, and number of nodes (eq. (1) and (2)) can be further improved, in terms of the specific network issues.

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

Amaral LAN. 2008. A truer measure of our ignorance. Proceedings of the National Academy of Sciences of USA, 105: 6795-6796

Barabasi AL, Albert R. 1999. Emergence of scaling in random networks. Science, 286(5439): 509

Barzel B, Barabási AL. 2013. Network link prediction by global silencing of indirect correlations. Nature

Biotechnology, 31: 720-725

Bastiaens P, Birtwistle MR, Blüthgen N, et al. 2015. Silence on the relevant literature and errors in implementation. Nature Biotechnology, 33: 336-339

Cancho RF, Sole RV. 2001. Optimization in Complex Networks. Santafe Institute, USA

Clauset A, Moore C, Newman MEJ. 2008. Hierarchical structure and the prediction of missing links in networks. Nature, 453: 98-101

Cohen JE, Briand F, Newman CM. 1990. Community Food Webs: Data and Theory. Springer, Berlin, Germany

Goemann B, Wingender E, Potapov AP. 2011. Topological peculiarities of mammalian networks with different functionalities: transcription, signal transduction and metabolic networks. Network Biology, 1(3-4): 134-148

Guimera R, Sales-Pardo M. 2009. Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences of USA, 106: 22073-22078

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

Lü LY, Medo M, Yeung CH, et al. 2012. Recommender systems. Physics Reports, 519: 1-49

Lü LY, Pan LM, Zhou T, et al. 2015. Toward link predictability of complex networks. Proceedings of the National Academy of Sciences of USA, 112: 2325-2330

Lü LY, Zhou T. 2011. Link prediction in complex networks: A survey. Physica A, 390: 1150-1170

Martinez ND. 1992. Constant connectance in community food webs. American Naturalist, 139: 1208-1218

Pathway Central. 2012. SABiosciences. http://www.sabiosciences.com/pathwaycentral.php

Rahman KMT, Md. Islam F, Banik RS, et al. 2013. Changes in protein interaction networks between normal and cancer conditions: Total chaos or ordered disorder? Network Biology, 3(1): 15-28

Shams B, Khansari M. 2014. Using network properties to evaluate targeted immunization algorithms. Network Biology, 4(3): 74-94

Yu HY, Braun P, Yildirim MA, et al. 2008. High-quality binary protein interaction map of the yeast interactome network. Science, 322: 104-110

Zhang WJ. 2007. Computer inference of network of ecological interactions from sampling data. Environmental Monitoring and Assessment, 124: 253-261

Zhang WJ. 2011. Constructing ecological interaction networks by correlation analysis: hints from community sampling. Network Biology, 1(2): 81-98

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

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

Zhang WJ. 2012c. Modeling community succession and assembly: A novel method for network evolution. Network Biology, 2(2): 69-78

Zhang WJ. 2013. Construction of Statistic Network from Field Sampling. In: Network Biology: Theories, Methods and Applications (WenJun Zhang, ed). 69-80, Nova Science Publishers, New York, USA

Zhang WJ, 2015a. A generalized network evolution model and self-organization theory on community assembly. Selforganizology, 2(3): 55-64

Zhang WJ. 2015b. A hierarchical method for finding interactions: Jointly using linear correlation and rank

correlation analysis. Network Biology, 5(4): 137-145

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

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

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

Zhang WJ. 2016b. Selforganizology: The Science of Self-Organization. World Scientific, Singapore

Zhang WJ, Li X. 2015. 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, Liu GH. 2012. Creating real network with expected degree distribution: A statistical simulation. Network Biology, 2(3): 110-117

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

Zhao J, Miao LL, Yang Y, et al. 2015. Prediction of links and weights in networks by reliable routes. Scientific Reports, 5: 12261

Zhou T. 2015. Why link prediction? http://blog.sciencenet.cn/blog-3075-912975.html. Accessed on Aug 14, 2015