Article

Creating real network with expected degree distribution: A statistical simulation

WenJun Zhang[1], GuangHua Liu[2]
[1]School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
[2]Guangzhou Experimental Station, Chinese Academy of Tropical Agricultural Sciences, Guangzhou 510140, China; Guangdong AIB Polytech College, Guangzhou 510507, China
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org, ghliu@gdaib.edu.cn

Received 16 December 2011; *Accepted 12 February 2012*; *Published online 1 September 2012*
IAEES

Abstract
The degree distribution of known networks is one of the focuses in network analysis. However, its inverse problem, i.e., to create network from known degree distribution has not yet been reported. In present study, a statistical simulation algorithm was developed to create real network with expected degree distribution. It is an iteration procedure in which a real network, with the least deviation of actual degree distribution to expected degree distribution, was created. Random assignment was used in the creation of connections. The Java program was designed. It may produce adjacency matrix, connection details, and actual degree distribution of the network created.

Keywords network; creation; degree distribution; statistical simulation.

1 Introduction

Based on either a probability distribution (e.g., Poisson distribution) or a frequency distribution, we may create a set of samples that follow the given distribution. The statistical characteristics of these theoretical distributions can be inferred and represented by these samples. It is obvious that the estimates from samples are either unbiased or biased. In present study, degree distribution means the frequency distribution of degree in a network. Degree distribution and connection structure of known networks is one of the focuses in network analysis (Dunne et al., 2002; Ibrahim et al., 2011; Paris and Bazzoni, 2011; Tacutu et al., 2011; Zhang, 2011; Zhang and Zhan, 2011). However, its inverse problem, i.e., to create network from known degree distribution has not yet been reported.

The present study aimed to present a statistical simulation algorithm that creates real network with expected degree distribution.

2 Algorithm

The statistical simulation procedures for creating real network with expected degree distribution are described bellow.

(1) Transform the expected degree distribution

$[d_1, d_2)$: f_1
$[d_2, d_3)$: f_2
…
$[d_m, d_{m+1})$: f_m

into (node, number of connections) pairs

n_1, s_1
n_2, s_2
…
n_p, s_p

where $[d_i, d_{i+1})$: interval of degree; f_i: number of nodes which degree falls in the interval $[d_i, d_{i+1})$; n_j: jth node; s_j: number of connections of jth node, and $p=\Sigma f_i$. There are totally Σs_i connections in the network.

At this step, each node falling in the same degree interval is randomly attributed an integer number (number of connections of this node) in this interval

$$s_j=(\text{int})((d_k-d_{k-1})*\text{Math.random}()+d_k).$$

(2) Rearrange p pairs of (node, number of connections), from larger to smaller in the number of connections

n_i, s_i
n_j, s_j
…
n_q, s_q

(3) For each node n_i with s_i expected connections, randomly create $n_i–m_i$ connections to other nodes and each of $n_i–m_i$ nodes has a connection, where m_i is the number of connections already created by previous nodes.

Given p constraints (p equalities for expected degree of p nodes), there will be $p(p-1)/2$ variables for connections. Therefore in general there are multiple connection plans, i.e. multiple networks may be obtained. Randomly creating connections to other nodes is a useful method. Moreover, the nodes with more connections should be handled in this procedure because of some reasons. For example, a reason is that the nodes with more connections are generally more important in the network.

(4) After all nodes are assigned connections, examine the symmetry of adjacency matrix, $A=(a_{ij})$, where $a_{ij}=1$ denotes there exists a connection between two nodes i and j, and $a_{ij}=0$ denotes there is not connection between nodes i and j. If $a_{ji}\neq a_{ij}$, then let $a_{ji}=a_{ij}=1$.

(5) If the deviation tolerance of degree distribution or the number of simulations is reached, terminate algorithm; or else return to (3). Here percent deviation of actual degree distribution to expected degree distribution is defined as

$$\Sigma|f_i - t_i|/p*100\%$$

where t_i: actual number of nodes in degree group *i*. If the number of simulations is reached, the results with the minimal percent deviation during simulation will be shown.

The following codes are the main Java algorithm, netConnect, used by Java applet netConnCreate (see http://www.iaees.org/publications/software/index.asp), to create a real network with expected degree distribution:

```
import java.io.*;
public class netConnect {
public int s,inn[],cols[][],netcon[][];
public double err;

public netConnect(int m, int in[], int ff[], double er, int sim) {
int i,j,k,l,u,v,c,cs,f[],p[],w[],innn[],colss[][];
double errr;
s=0;
for(i=1;i<=m;i++) s+=ff[i];
f=new int[s+1];
p=new int[s+1];
w=new int[s+1];
inn=new int[m+1];
innn=new int[m+1];
cols=new int[s+1][s+1];
colss=new int[s+1][s+1];
k=u=0;
for(i=1;i<=m;i++)
for(j=1;j<=ff[i];j++) {
k++;
f[k]=(int)((in[i+1]-in[i])*Math.random()+in[i]);
u+=f[k]; }
netcon=new int[u+1][u+1];
for(i=1;i<=s;i++) p[i]=i;
for(i=1;i<=s-1;i++)    {
k=i;
for(j=i;j<=s-1;j++)
if(f[j+1]>f[k])    k=j+1;
l=p[i];
p[i]=p[k];
p[k]=l;
u=f[i];
f[i]=f[k];
f[k]=u; }
k=0;
err=1e+50;
do {
for(i=1;i<=s;i++)
for(j=1;j<=s;j++) colss[i][j]=0;
loop: for(i=1;i<=s;i++)    {
v=0;
for(j=1;j<=i-1;j++)
if (colss[j][i]==1) {
colss[i][j]=1;
v++;
if (v==f[i]) continue loop; }
c=0;
for(j=1;j<=s;j++) w[j]=j;
while (f[i]!=0) {
cs=(int)((s-c)*Math.random()+1);
if (w[cs]==i) continue;
if ((colss[w[cs]][i]==0) & (w[cs]<i)) continue;
colss[i][w[cs]]=1;
if (cs<s-c) for(j=cs+1;j<=s-c;j++) w[j-1]=w[j];
c++;
if (c>=(f[i]-v)) break; } }
for(i=1;i<=s;i++)
for(j=1;j<=s;j++)
if (colss[j][i]!=colss[i][j]) colss[j][i]=colss[i][j]=1;
for(i=1;i<=s;i++) {
```

```
p[i]=0;
for(j=1;j<=s;j++)
if (colss[i][j]==1) p[i]++; }
for(i=1;i<=m;i++) {
innn[i]=0;
for(j=1;j<=s;j++)
if ((p[j]>=in[i]) & (p[j]<in[i+1])) innn[i]++; }
errr=0;
for(i=1;i<=m;i++) errr+=Math.abs(ff[i]-innn[i]);
errr/=s;
if (errr<err) {
err=errr;
for(i=1;i<=s;i++)
for(j=1;j<=s;j++) cols[i][j]=colss[i][j];
for(i=1;i<=m;i++) inn[i]=innn[i]; }
k++;
if (k>sim) break;
} while(errr>er);
System.out.println("\nAdjacency matrix:");
for(i=1;i<=s;i++) {
for(j=1;j<=s;j++)
System.out.print(String.valueOf(cols[i][j])+"    ");
System.out.println(); }
System.out.println("\nNode      "+"to Node"+"       Connection");
for(i=1;i<=s;i++)
for(j=1;j<=s;j++)
if (cols[i][j]==1) {
System.out.print(String.valueOf(i)+"                "+String.valueOf(j)+"                  1"+"\n");
netcon[i][j]=1; }
System.out.println("\nExpected degree distribution of network:");
for(i=1;i<=m;i++)
System.out.println("["+in[i]+", "+in[i+1]+"): "+ff[i]);
System.out.println();
System.out.println("\nActual degree distribution of created network:");
for(i=1;i<=m;i++)
System.out.println("["+in[i]+", "+in[i+1]+"): "+inn[i]);
System.out.print("\nPercent deviation of actual degree distribution to expected degree distribution:
"+String.valueOf((int)(err*10000)/100.00)+"%\n");
} }
```

3 Results

Suppose there is an expected degree distribution for the network, in which there are nine degree groups

[1, 3): 1
[3, 5): 2
[5, 7): 4
[7, 9): 8
[9, 11): 5
[11, 13): 3
[13, 15): 2
[15, 17): 2
[17, 19): 1

In the data file for the algorithm, the expected degree distribution should be stored in the following format

1 2 4 8 5 3 2 2 1
1 3 5 7 9 11 13 15 17 19

Running the algorithm above, the parameter input interface of Java algorithm is shown as in Fig. 1

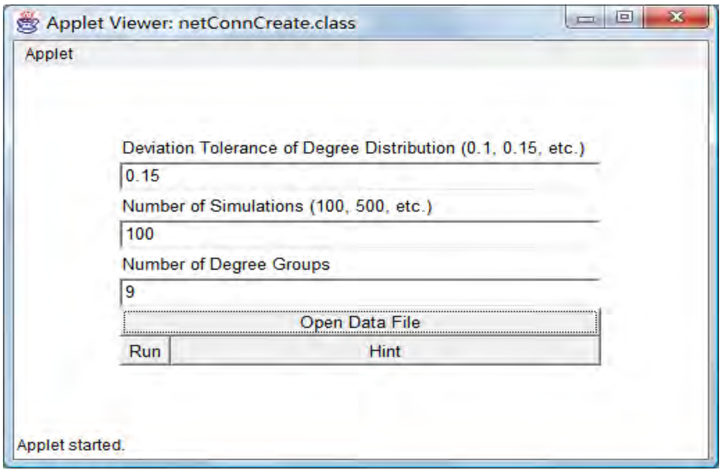

Fig. 1 Interface of Java algorithm

The adjacency matrix was produced as the following

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

Table 1 Connection details of the real network created

FN	TN	Conn	FN	TN	Conn	FN	TN	Conn	FN	TN	Conn	FN	TN	Conn
1	2	1	4	11	1	8	20	1	14	19	1	20	15	1
1	5	1	4	13	1	8	22	1	14	22	1	21	1	1
1	6	1	4	15	1	8	26	1	14	28	1	21	3	1
1	10	1	4	16	1	8	28	1	15	2	1	21	12	1
1	11	1	4	17	1	9	5	1	15	4	1	21	15	1
1	12	1	4	18	1	9	6	1	15	8	1	21	27	1
1	14	1	4	20	1	9	12	1	15	18	1	22	1	1
1	16	1	4	22	1	9	16	1	15	19	1	22	2	1
1	17	1	4	28	1	9	17	1	15	20	1	22	3	1
1	18	1	5	1	1	9	18	1	15	21	1	22	4	1
1	19	1	5	2	1	9	19	1	15	23	1	22	5	1
1	20	1	5	3	1	9	23	1	16	1	1	22	6	1
1	21	1	5	8	1	9	24	1	16	3	1	22	7	1
1	22	1	5	9	1	10	1	1	16	4	1	22	8	1
1	23	1	5	11	1	10	2	1	16	6	1	22	13	1
1	25	1	5	12	1	10	3	1	16	9	1	22	14	1
1	26	1	5	17	1	10	4	1	16	24	1	23	1	1
2	1	1	5	18	1	10	6	1	16	27	1	23	2	1
2	3	1	5	22	1	10	8	1	16	28	1	23	9	1
2	4	1	5	26	1	10	19	1	17	1	1	23	15	1
2	5	1	6	1	1	10	28	1	17	2	1	23	24	1
2	6	1	6	2	1	11	1	1	17	4	1	23	28	1
2	10	1	6	4	1	11	3	1	17	5	1	24	3	1
2	13	1	6	9	1	11	4	1	17	7	1	24	6	1
2	14	1	6	10	1	11	5	1	17	9	1	24	7	1
2	15	1	6	12	1	11	7	1	18	1	1	24	9	1
2	17	1	6	13	1	11	14	1	18	3	1	24	16	1
2	19	1	6	16	1	11	18	1	18	4	1	24	23	1
2	20	1	6	19	1	11	28	1	18	5	1	25	1	1
2	22	1	6	20	1	12	1	1	18	7	1	25	7	1
2	23	1	6	22	1	12	5	1	18	9	1	25	13	1
2	28	1	6	24	1	12	6	1	18	11	1	25	27	1
3	2	1	7	3	1	12	7	1	18	12	1	26	1	1
3	5	1	7	4	1	12	9	1	18	13	1	26	5	1
3	7	1	7	11	1	12	18	1	18	15	1	26	7	1
3	8	1	7	12	1	12	21	1	19	1	1	26	8	1
3	10	1	7	17	1	12	28	1	19	2	1	27	3	1
3	11	1	7	18	1	13	2	1	19	6	1	27	16	1
3	14	1	7	19	1	13	4	1	19	7	1	27	21	1
3	16	1	7	22	1	13	6	1	19	8	1	27	25	1
3	18	1	7	24	1	13	8	1	19	9	1	28	2	1
3	21	1	7	25	1	13	18	1	19	10	1	28	3	1
3	22	1	7	26	1	13	20	1	19	14	1	28	4	1
3	24	1	8	3	1	13	22	1	19	15	1	28	8	1
3	27	1	8	5	1	13	25	1	20	1	1	28	10	1
3	28	1	8	10	1	14	1	1	20	2	1	28	11	1
4	2	1	8	13	1	14	2	1	20	4	1	28	12	1
4	6	1	8	14	1	14	3	1	20	6	1	28	14	1
4	7	1	8	15	1	14	8	1	20	8	1	28	16	1
4	10	1	8	19	1	14	11	1	20	13	1	28	23	1

FN: from node; TN: to node; Conn: 1's mean undirected connections.

The connection details of the real network created are indicated in Table 1. It should be noted that connection details in Table 1 can be easily transformed into adjacency matrix. The connection details are equivalent to the adjacency matrix. The actual degree distribution of created network is

[1, 3): 0
[3, 5): 3
[5, 7): 4
[7, 9): 8
[9, 11): 5
[11, 13): 4
[13, 15): 2
[15, 17): 1
[17, 19): 1

Percent deviation of actual degree distribution to expected degree distribution is 14.28%. Thus the network created is a better realization of expected network.

Using the data of connection details above, I draw a network using the software (Zhang, 2012), as indicated in Fig. 2.

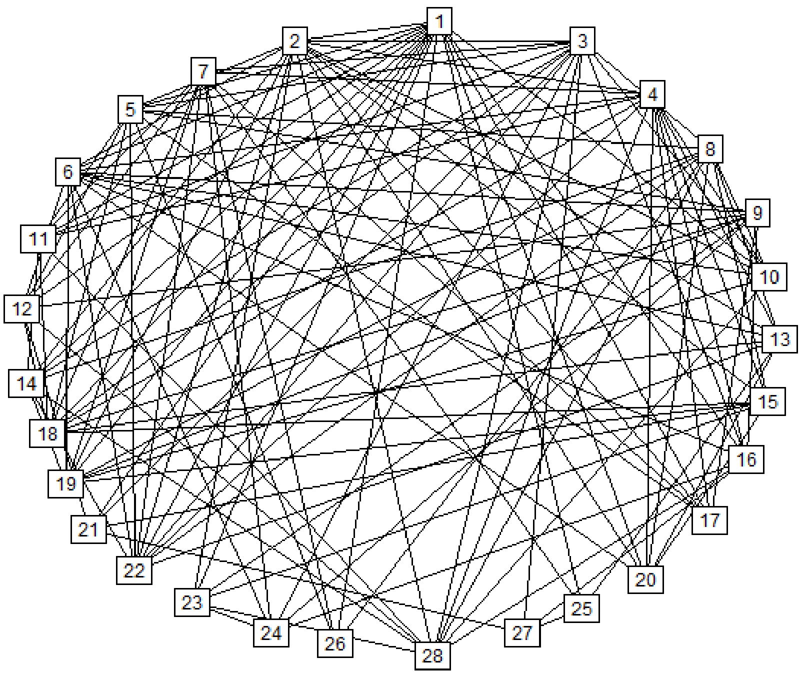

Fig. 2 A real network created by the algorithm

4 Discussion

In the algorithm, if allowed number of simulations reached but the percent deviation of actual degree distribution to the expected degree distribution is still large, we may increase the number of simulations and restart the algorithm. The tolerance deviation should not be too small in the algorithm.

In addition to the algorithm above, other methods can also be used to create real network with expected degree distribution. For example, connections may be obtained by using optimization techniques. But for large networks the computation will be time-consuming.

References

Dunne JA, Williams RJ, Martinez ND. 2002. Food-web structure and network theory: the role of connectance and size. Ecology, 99(20): 12917-12922

Ibrahim SS, Eldeeb MAR, Rady MAH. 2011. The role of protein interaction domains in the human cancer network. Network Biology, 1(1): 59-71

Paris L, Bazzoni G. 2011. The polarity sub-network in the yeast network of protein-protein interactions. Network Biology, 1(3-4): 149-158

Tacutu R, Budovsky A, Yanai H, et al. 2011.Immunoregulatory network and cancer-associated genes: molecular links and relevance to aging. Network Biology, 1(2): 112-120

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

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

Zhang WJ. 2012. A Java software for drawing graphs. Network Biology, 2(1): 38-44