Article

netAna: A tool for network analysis

WenJun Zhang

School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China

E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 18 July 2023; *Accepted 3 August 2023*; *Published online 10 August 2023*; *Published 1 December 2023*

Abstract

In present study, a software tool for network analysis is presented. The tool is mainly used to calculate topological properties of the network, detect type of the network, and calculate node centralities of the network. Various network properties include total number of nodes, total number of links, clustering coefficient of network, mean link density of network, connectance of network, skewness of degree distribution, aggregation index of network, variation coefficient of network, entropy of network, etc. Binomial distribution, Poisson distribution, exponential distribution, and particularly power law distribution are used for detection of network type. Node centralities used include degree centrality, closeness centrality, and betweenness centrality.

Keywords network analysis; software; network topology; power law; node centrality; crucial nodes.

Network Biology
ISSN 2220-8879
URL: http://www.iaees.org/publications/journals/nb/online-version.asp
RSS: http://www.iaees.org/publications/journals/nb/rss.xml
E-mail: networkbiology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Network analysis is one of the research focuses in network biology (Zhang, 2012, 2016g, 2018). In the past, various algorithms and tools on network analysis have been proposed and used in network biology (Jiang and Zhang, 2015; Jiang et al., 2015; Chen and Zhang, 2020; Xing and Zhang, 2020, 2021; Yang and Zhang, 2022; Zhang and Zhang, 2019; Zhang and Feng, 2017; Zhang and Qi, 2023; Zhang, 2012, 2016a-e, 2017a-b, 2018a-b, 2021, 2023). In present study, I will use Delphi developing environment (Zhang, 2023) to develop a standalone executable software tool for network analysis. It is expected to be powerful and easily used.

2 Algorithms and Software

2.1 Algorithms

All algorithms for detection of network type and calculation of network properties and node centralities, are based on Zhang and Zhan (2011), and Zhang (2016f), etc.

Various network properties include total number of nodes, total number of links, clustering coefficient of network, mean link density of network, connectance of network, skewness of degree distribution, aggregation index of network, variation coefficient of network, entropy of network (Zhang and Zhang, 2011; Zhang, 2012, 2016g, 2018). Binomial distribution, Poisson distribution, exponential distribution ($F(x)$=1-e$^{-\lambda x}$, x≥0), particularly power law distribution ($p(x)=x^{-\alpha}$, $x \geq x_{\min}$) are used for detection of network type.

Node centralities used in present study include degree centrality, closeness centrality, and betweenness centrality (Zhang, 2012, 2016f, 2018).

2.2 Software codes

The full Object Pascal codes of netAna (version 1.0), developed in Delphi environment, are as follows:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Math;

type
  TForm1 = class(TForm)
    OpenDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
    procedure Rankings();
    procedure Dijkstra();
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  nname: array[1..50000,1..3] of String;
  nodes: array[1..10000] of String;
  d: array of array of integer;
  distances: array of array of Double;
  centr: array of Extended;
  pat, rank: array of Integer;
  sss: Extended;
  v: integer;

implementation

{$R *.dfm}

procedure TForm1.Rankings();
var
```

```
     s, i, j, k: Integer;
     u: Double;
begin
setLength(rank,v+1);
for i:=1 to v do
rank[i]:=i;
for i:=1 to v-1 do
begin
k:=i;
for j:=i to v-1 do
if(centr[j+1]>centr[k]) then
k:=j+1;
s:=rank[i];
rank[i]:=rank[k];
rank[k]:=s;
u:=centr[i];
centr[i]:=centr[k];
centr[k]:=u;
end;
end;

procedure TForm1.Dijkstra();
var
     a, b, p, w: array of Integer;
     n, h, su, c, sd, ma, iv, i, j, k: integer;
begin
//d: weighted adjacency matrix; ss: total number of paths;
//pat: number of paths passing through each node;
//distances: matrix of distances between different nodes;
//paths: string of paths and distances between any of two nodes.
setLength(distances,v+1,v+1);
setLength(pat,v+1);
setLength(a,v+1);
setLength(b,v+1);
setLength(p,v+1);
setLength(w,v+1);
for i:=1 to v do
for j:=1 to v do
if ((d[i,j]=0) and (i<>j)) then
d[i,j]:=12345;
su:=0;
for j:=1 to v-1 do
begin
for k:=j+1 to v do
begin
for i:=1 to v do
begin
p[i]:=0;
w[i]:=0;
a[i]:=12345;
```

```
end;
a[j]:=0;
w[j]:=1;
n:=j;
h:=0;
while(v>0) do
begin
ma:=12345;
for i:=1 to v do
begin
if(w[i]=1) then
continue;
iv:=d[n,i]+a[n];
if(iv<a[i]) then
begin
a[i]:=iv;
b[i]:=n;
end;
if(a[i]>ma) then
continue;
ma:=a[i];
h:=i;
end;
w[h]:=1;
if(h=k) then
break;
n:=h;
end;
sd:=a[k];
p[1]:=k;
c:=k;
for i:=2 to v do
begin
if(c=j) then
break;
p[i]:=b[c];
c:=b[c];
end;
for i:=v downto 1 do
for h:=1 to v do
if(p[i]=h) then
begin
pat[h]:=pat[h]+1;
break;
end;
if(sd<>12345) then
begin
distances[j,k]:=sd;
distances[k,j]:=sd;
end;
```

```
if(sd=12345) then
su:=su+1;
end;
end;
for i:=1 to v do
for j:=1 to v do
if ((d[i,j]=12345) and (i<>j)) then
d[i,j]:=0;
sss:=v*(v-1)/2-su;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
Application.terminate;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
if SaveDialog1.Execute then
Memo1.Lines.SaveToFile(SaveDialog1.FileName+'.txt');
Application.terminate;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
   F: TextFile;
   filename, str: String;
   n, m, r, rr, rexp, i, j, k, tem, xm, xmin: Integer;
   it, pp, qq, ss, meann, varr, skew, h, pr0, hk, chi, k1, k2: Extended;
   a, alpha, alpha1, L, dd: Extended;
   deg, frq, x, xminss, xmins, z: array of Integer;
   p, pr, fr, dat, cx, cf: array of Single;
begin
Form1.visible:=false;
try
if OpenDialog1.Execute then
begin
filename:=OpenDialog1.Filename;
AssignFile(F,filename);
Reset(F);
n:=0;
while not Eof(F) do
begin
n:=n+1;
Readln(F,str);
i:=pos(',', str);
nname[n,1]:=copy(str,1,i-1);
str:=copy(str,i+1,length(str)-i);
i:=pos(',', str);
nname[n,2]:=copy(str,1,i-1);
```

```
nname[n,3]:=copy(str,i+1,length(str)-i);
nname[n,1]:=trim(nname[n,1]);
nname[n,2]:=trim(nname[n,2]);
nname[n,3]:=trim(nname[n,3]);
end;
CloseFile(F);
end;
v:=1;
nodes[1]:=nname[1,1];
for i:=1 to n do
for k:=1 to 2 do
begin
tem:=0;
for j:=1 to v do
if(sametext(nodes[j],nname[i,k])) then
begin
tem:=1;
break;
end;
if(tem=0) then
begin
v:=v+1;
nodes[v]:=nname[i,k];
end;
end;
memo1.Lines.Append('');
memo1.Lines.Append('I. Global properties of network');
memo1.Lines.Append('');
memo1.Lines.append('Total number of nodes='+inttostr(v));
memo1.Lines.append('Total number of links='+inttostr(n));
memo1.Lines.append('');
setLength(d,v+1,v+1);
for tem:=1 to n do
begin
for k:=1 to v do
if(sametext(nodes[k],nname[tem,1])) then
begin
i:=k;
break;
end;
for k:=1 to v do
if(sametext(nodes[k],nname[tem,2])) then
begin
j:=k;
break;
end;
d[i,j]:=1;
d[j,i]:=1;
end;
//Network type detection
```

```
rr:=10;
setLength(deg,v+1);
setLength(centr,v+1);
setLength(frq,rr+1);
a:=0;
for i:=1 to v do
begin
deg[i]:=0;
for j:=1 to v do
begin
if (abs(d[i,j])=1) then
deg[i]:=deg[i]+1;
end;
a:=a+deg[i];
centr[i]:=deg[i];
end;
memo1.Lines.append('Clustering coefficient of network: '+floattostr(a/v));
memo1.Lines.append('Mean link density of network: '+floattostr(n/v));
memo1.Lines.append('Connectance of network: '+floattostr(2*n/(v*v-v)));
memo1.Lines.Append('');
for i:=v downto 1 do
for j:=1 to v-1 do
if(deg[j]<deg[j+1]) then
begin
tem:=deg[j];
deg[j]:=deg[j+1];
deg[j+1]:=tem;
end;

pp:=0;

qq:=0;
for i:=1 to v do
begin
pp:=pp+deg[i];
qq:=qq+deg[i]*(deg[i]-1);
end;
it:=(deg[1]-deg[v])/rr;
for i:=1 to rr do
begin
frq[i]:=0;
for j:=1 to v do
begin
if((deg[j]>=(deg[v]+(i-1)*it)) and (deg[j]<(deg[v]+i*it))) then
frq[i]:=frq[i]+1;
end;
end;
rexp:=rr;
rr:=0;
for i:=1 to rexp do
```

```
if(frq[i]<>0) then
rr:=rr+1;
j:=0;
for i:=1 to rr do
begin
if(frq[i]<>0) then
j:=j+1;
end;
setLength(fr,j+1);
j:=0;
for i:=1 to rr do
begin
if(frq[i]<>0) then
begin
j:=j+1;
fr[j]:=frq[i];
end;
end;
memo1.Lines.Append('II. Network properties and network type detection');
memo1.Lines.Append(");
meann:=0;
for i:=1 to v do
meann:=meann+deg[i];
meann:=meann/v;
varr:=0;
for i:=1 to v do
varr:=varr+power(deg[i]-meann,2);
varr:=varr/(v-1);
skew:=v/((v-2)*sqrt(varr));
memo1.Lines.Append('Skewness of degree distribution: '+floattostr(skew)+");
memo1.Lines.Append(");
h:=v*qq/(pp*(pp-1));
memo1.Lines.Append('Aggregation index of network: '+floattostr(h)+");
if(h<=1) then
memo1.Lines.Append('It is likely a random network.')
else memo1.Lines.Append('It is likely a complex network.');
memo1.Lines.Append(");
h:=varr/meann;
memo1.Lines.Append('Variation coefficient H of network: '+floattostr(h)+");
memo1.Lines.Append('Entropy E of network: '+floattostr(varr-meann)+");
if(h<=1) then
memo1.Lines.Append('It is likely a random network.')
else memo1.Lines.Append('It is likely a complex network.');
memo1.Lines.Append(");
setLength(p,rr);
setLength(pr,rr);
//Binomial distri.: pr:= Crn pr qn-r, r:=0,1,2,…, n;
ss:=0;
for i:=1 to rr do
ss:=ss+fr[i]*(i-1);
```

```
pp:=ss/(v*(rr-1));
qq:=1-pp;
pr0:=power(qq,rr-1);
for i:=1 to rr-1 do
begin
if(i=1) then
pr[i]:=(rr-i)*pp*pr0/(i*qq)
else pr[i]:=(rr-i)*pp*pr[i-1]/(i*qq);
end;
chi:=0;
for i:=0 to rr-1 do
begin
if(i=0) then
hk:=pr0*v
else hk:=pr[i]*v;
if((i=0) and (pr0=0)) then
hk:=fr[i+1];
if((i>0) and (p[i]=0)) then
hk:=fr[i+1];
if((i=0) and (1/hk>5)) then
chi:=chi+power(pr0*v-fr[i+1],2)/hk
else chi:=chi+power(pr[i]*v-fr[i+1],2)/hk;
end;
memo1.Lines.Append('Binomial distribution Chi-square:='+floattostr(chi)+");
memo1.Lines.Append('Binomial p:='+floattostr(pp)+");
k1:=20.09;
if (chi<=k1) then
begin
memo1.Lines.Append('Node degrees are binomially distributed. ');
memo1.Lines.Append('It is likely a random network. ');
end
else
memo1.Lines.Append('It is likely not a random network.');
memo1.Lines.Append(");
//Poisson distri., pr := e^(-λ)λr/r! , r:=0,1,2,…
pr0:=exp(-meann);
for r:=1 to rr-1 do
begin
if(r=1) then
pr[r]:=meann/r*pr0
else pr[r]:=meann/r*pr[r-1];
end;
chi:=0;
for i:=0 to rr-1 do
begin
if(i=0) then
hk:=pr0*v
else hk:=pr[i]*v;
if((i=0) and (pr0=0)) then
hk:=fr[i+1];
```

```
if((i>0) and (p[i]=0)) then
hk:=fr[i+1];
if((i=0) and (1/hk>5)) then
chi:=chi+power(pr0*v-fr[i+1],2)/hk
else chi:=chi+power(pr[i]*v-fr[i+1],2)/hk;
end;
memo1.Lines.Append('Poisson distribution Chi-square:='+floattostr(chi)+");
memo1.Lines.Append('Poisson distribution λ:='+floattostr(meann)+");
k1:=20.09;
if(chi<=k1) then
begin
memo1.Lines.Append('Node degrees are Poisson distributed. ');
memo1.Lines.Append('It is likely a random network.');
end
else memo1.Lines.Append('It is likely not a random network.');
memo1.Lines.Append(");
//Exponential distri., F(x) =1-e^(-λx), x≥0
chi:=0;
for i:=1 to rexp do
begin
if(frq[i]<>0) then
begin
k1:=deg[v]+it/2+(i-1)*it;
k2:=deg[v]+it/2+i*it;
pp:=v*(exp(-k1/meann)-exp(-k2/meann));
chi:=chi+power(frq[i]-pp,2)/pp;
end;
end;
memo1.Lines.Append('Exponential distribution Chi-square:='+floattostr(chi)+");
memo1.Lines.Append('Exponential distribution λ (Exponential distri.: F(x)=1-e^(-λx), x≥0; x: node degree):='+floattostr(1/meann)+");
k1:=20.09;
if(chi<=k1) then
memo1.Lines.Append('Node degrees are exponentially distributed.')
else
memo1.Lines.Append('Node degrees are not exponentially distributed.');
memo1.Lines.Append(");
//Power law distri., p(x):=x^(-α), x≥xmin
setLength(x,v+1);
for i:=1 to v do
x[i]:=deg[i];
setLength(xminss,v+1);
m:=1;
xminss[1]:=x[1];
for i:=2 to v do
begin
tem:=0;
for j:=1 to m do
if(xminss[j]=x[i]) then
begin
```

```
tem:=1;
break;
end;
if(tem=0) then
begin
m:=m+1;
xminss[m]:=x[i];
end;
end;
setLength(xmins,m);
for i:=1 to m do
xmins[i]:=xminss[m-i+1];
setLength(dat,m);
setLength(z,v+1);
for i:=1 to v do
z[i]:=deg[v-i+1];
n:=v;
for xm:=1 to m-1 do
begin
xmin:=xmins[xm];
j:=0;
for i:=1 to n do
if(z[i]>=xmin) then
begin
j:=j+1;
z[j]:=z[i];
end;
n:=j;
a:=0;
for j:=1 to n do
a:=a+ln(z[j]/xmin);
a:=n/a;
setLength(cx,n+1);
setLength(cf,n+1);
for j:=1 to n do
cx[j]:=(j-1)/n;
for j:=1 to n do
cf[j]:=1-power(xmin/z[j],a);
dat[xm]:=-1e+20;
for j:=1 to n do
if(abs(cf[j]-cx[j])>dat[xm]) then
dat[xm]:=abs(cf[j]-cx[j]);
end;
dd:=1e+20;
for j:=1 to m-1 do
if(dat[j]<dd) then
dd:=dat[j];
for i:=1 to m-1 do
if(dat[i]<=dd) then
begin
```

```
xmin:=xmins[i];
break;
end;
j:=0;
for i:=1 to v do
if(x[i]>=xmin) then
j:=j+1;
setLength(z,j+1);
j:=0;
for i:=1 to v do
if(x[i]>=xmin) then
begin
j:=j+1;
z[j]:=x[i];
end;
n:=j;
alpha1:=0;
for i:=1 to n do
alpha1:=alpha1+ln(z[i]/xmin);
alpha:=1+n/alpha1;
L:=n*ln((alpha-1)/xmin)-alpha*alpha1;
//L is the log-likelihood of the data x>:=xmin under the fitted power law.
memo1.Lines.Append('Power law distribution Kolmogorov-Smirnov goodness-of-fit statistic D:='+floattostr(dd)+");
if(dd<(1.63/sqrt(n))) then
memo1.Lines.Append('Node degrees are power-law distributed (Power law distri.: p(x)=x^(-α), x≥xmin; x: node degree), thus it is a scale-free complex network.');
memo1.Lines.Append('Power law α:='+floattostr(alpha)+");
memo1.Lines.Append('Power law xmin:='+floattostr(xmin)+");
memo1.Lines.Append(");
memo1.Lines.Append('III. Node centralities');
memo1.Lines.Append(");
//Node centrality calculation
memo1.Lines.Append('Degree centrality');
memo1.Lines.Append(");
memo1.Lines.Append('Rank'+'        '+'Node'+'        '+'Degree centrality');
Rankings();
for i:=1 to v do
memo1.Lines.Append(inttostr(i)+'        '+nodes[rank[i]]+'        '+floattostr(centr[i]));
memo1.Lines.Append(");
memo1.Lines.Append('In terms of degree centrality, '+nodes[rank[1]]+' and '+nodes[rank[2]]+', etc., are the most crucial nodes in the network.');
memo1.Lines.Append('And '+nodes[rank[v]]+' and '+nodes[rank[v-1]]+', etc., are the most trivial nodes in the network.');
memo1.Lines.Append(");
memo1.Lines.Append('Closeness centrality');
memo1.Lines.Append(");
memo1.Lines.Append('Rank'+'        '+'Node'+'        '+'Closeness centrality');
Dijkstra();
for i:=1 to v do
begin
ss:=0;
```

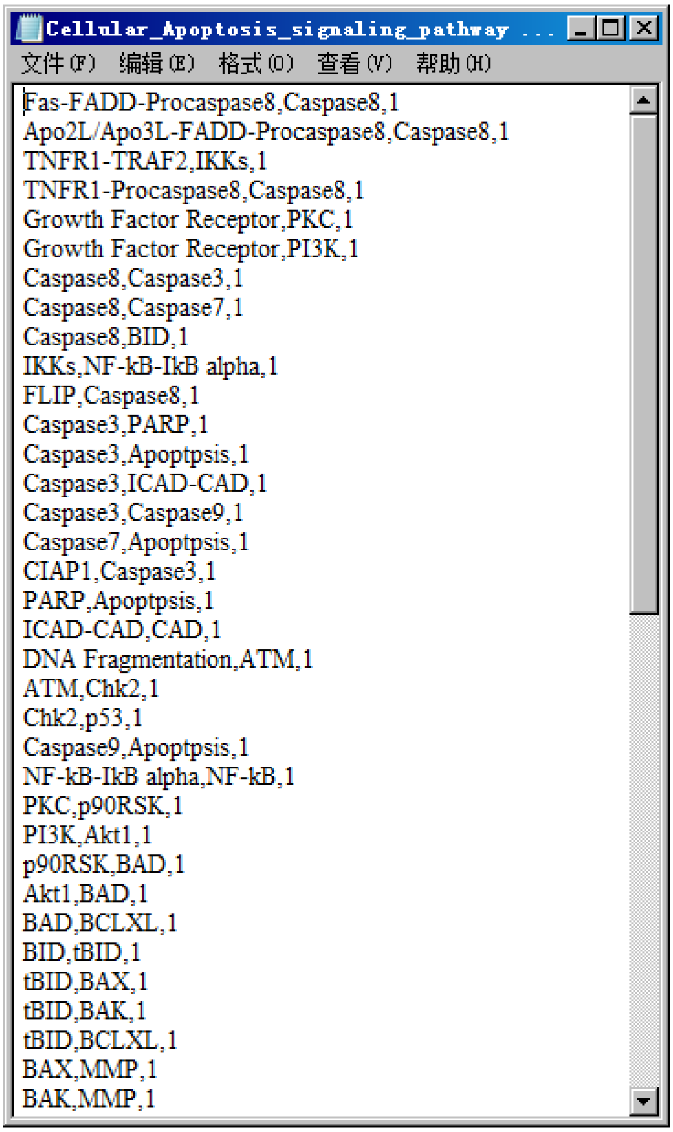
```
for j:=1 to v do
ss:=ss+distances[i,j];
centr[i]:=1/ss;
end;
Rankings();
for i:=1 to v do
memo1.Lines.Append(inttostr(i)+'        '+nodes[rank[i]]+'        '+floattostr(centr[i]));
memo1.Lines.Append('');
memo1.Lines.Append('In terms of closeness centrality, '+nodes[rank[1]]+' and '+nodes[rank[2]]+', etc., are the most crucial nodes in the network.');
memo1.Lines.Append('And '+nodes[rank[v]]+' and '+nodes[rank[v-1]]+', etc., are the most trivial nodes in the network.');
memo1.Lines.Append('');
memo1.Lines.Append('Betweenness centrality');
memo1.Lines.Append('');
memo1.Lines.Append('Rank'+'        '+'Node'+'        '+'Betweenness centrality');
for i:=1 to v do
centr[i]:=pat[i]/sss;
Rankings();
for i:=1 to v do
memo1.Lines.Append(inttostr(i)+'        '+nodes[rank[i]]+'        '+floattostr(centr[i]));
memo1.Lines.Append('');
memo1.Lines.Append('In terms of betweeness centrality, '+nodes[rank[1]]+' and '+nodes[rank[2]]+', etc., are the most crucial nodes in the network.');
memo1.Lines.Append('And '+nodes[rank[v]]+' and '+nodes[rank[v-1]]+', etc., are the most trivial nodes in the network.');
memo1.Lines.Append('');
except
exit;
end;
end;

end.
```

2.3 Data

The data for the network is stored in a plain text file. Suppose there are totally *m* links in the network, then there are *m* rows in the data file. There are three items in each row: the first item (node) and the second item (node) represent a link between two nodes and the third item is an integer 1. The three items are separated by the comma “,” (so by default, it means that any comma is not allowed in the first and second items).

The software and demo data files are included in the package: http://www.iaees.org/publications/journals/nb/articles/2023-13(4)/e-suppl/Zhang-Supplementary-Material3.rar


Cellular_Apoptosis_signaling_pathway ...
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
Fas-FADD-Procaspase8,Caspase8,1
Apo2L/Apo3L-FADD-Procaspase8,Caspase8,1
TNFR1-TRAF2,IKKs,1
TNFR1-Procaspase8,Caspase8,1
Growth Factor Receptor,PKC,1
Growth Factor Receptor,PI3K,1
Caspase8,Caspase3,1
Caspase8,Caspase7,1
Caspase8,BID,1
IKKs,NF-kB-IkB alpha,1
FLIP,Caspase8,1
Caspase3,PARP,1
Caspase3,Apoptpsis,1
Caspase3,ICAD-CAD,1
Caspase3,Caspase9,1
Caspase7,Apoptpsis,1
CIAP1,Caspase3,1
PARP,Apoptpsis,1
ICAD-CAD,CAD,1
DNA Fragmentation,ATM,1
ATM,Chk2,1
Chk2,p53,1
Caspase9,Apoptpsis,1
NF-kB-IkB alpha,NF-kB,1
PKC,p90RSK,1
PI3K,Akt1,1
p90RSK,BAD,1
Akt1,BAD,1
BAD,BCLXL,1
BID,tBID,1
tBID,BAX,1
tBID,BAK,1
tBID,BCLXL,1
BAX,MMP,1
BAK,MMP,1
```

Fig. 2 A data file for netAna.

3 Demonstration

The network data for cellular apoptosis signaling pathway were obtained from Huang and Zhang (2012), Li and Zhang (2013), which was stored in a text file (Cellular_Apoptosis_signaling_pathway.txt), as shown in Fig. 2.

Running netAna and loading the data file, the results (Fig. 3) are listed bellow:

```
I. Global properties of network

Total number of nodes=49
Total number of links=67

Clustering coefficient of network: 2.73469387755102
Mean link density of network: 1.36734693877551
Connectance of network: 0.0569727891156463
```

```
II. Network properties and network type detection

Skewness of degree distribution: 0.53606751912101

Aggregation index of network: 1.13825608798115
It is likely a complex network.

Variation coefficient H of network: 1.38308457711443
Entropy E of network: 1.04761904761905
It is likely a complex network.

Binomial distribution Chi-square:=47.7728601257816
Binomial p:=0.255102040816327
It is likely not a random network.

Poisson distribution Chi-square:=105.214297758277
Poisson distribution λ:=2.73469387755102
It is likely not a random network.

Exponential distribution Chi-square:=65.7383685892767
Exponential distribution λ(Exponential distri.: F(x)=1-e^(-λx), x≥0; x: node degree):=0.365671641791045
Node degrees are not exponentially distributed.

Power law distribution Kolmogorov-Smirnov goodness-of-fit statistic D:=0.214274525642395
Node degrees are power-law distributed (Power law distri.: p(x)=x^(-α), x≥xmin; x: node degree), thus it is a scale-free complex network.
Power law α:=3.27938463678571
Power law xmin:=3

III. Node centralities

Degree centrality

Rank      Node      Degree centrality
1      MMP      11
2      Caspase8      7
3      Caspase3      6
4      Apoptpsis      6
5      BAK      6
6      BCL2      6
7      p53      5
8      CIAP1      4
9      DNA Fragmentation      4
10      BCLXL      4
11      tBID      4
```

12	Caspase9	3
13	BAD	3
14	BAX	3
15	BIM	3
16	ICAD-CAD	2
17	BID	2
18	PKC	2
19	CAD	2
20	PI3K	2
21	ATM	2
22	Chk2	2
23	Growth Factor Receptor	2
24	p90RSK	2
25	Akt1	2
26	NF-kB-IkB alpha	2
27	Caspase7	2
28	IKKs	2
29	VDAC	2
30	PARP	2
31	PUMA	2
32	Noxa	2
33	SMAC	2
34	Arts	2
35	HTRA2	2
36	Cytoc	2
37	EndoG	2
38	AIF	2
39	APAF1	2
40	MDM2	2
41	TNFR1-Procaspase8	1
42	NF-kB	1
43	FLIP	1
44	TNFR1-TRAF2	1
45	Fas-FADD-Procaspase8	1
46	p53AIP1	1
47	p14(ARF)	1
48	Apo2L/Apo3L-FADD-Procaspase8	1
49	POD	1

In terms of degree centrality, MMP and Caspase8, etc., are the most crucial nodes in the network.
And POD and Apo2L/Apo3L-FADD-Procaspase8, etc., are the most trivial nodes in the network.

Closeness centrality

Rank	Node	Closeness centrality

1	IKKs	0.25
2	NF-kB-IkB alpha	0.25
3	TNFR1-TRAF2	0.166666666666667
4	NF-kB	0.166666666666667
5	MMP	0.0091743119266055
6	BAK	0.00806451612903226
7	Apoptpsis	0.00793650793650794
8	BCLXL	0.0078125
9	BCL2	0.00763358778625954
10	Caspase3	0.00709219858156028
11	BAX	0.00709219858156028
12	tBID	0.00699300699300699
13	SMAC	0.00699300699300699
14	Arts	0.00699300699300699
15	HTRA2	0.00699300699300699
16	EndoG	0.00699300699300699
17	AIF	0.00699300699300699
18	Cytoc	0.00671140939597315
19	CIAP1	0.00641025641025641
20	Caspase7	0.00641025641025641
21	Caspase9	0.00641025641025641
22	BIM	0.00636942675159236
23	BID	0.00628930817610063
24	Caspase8	0.00628930817610063
25	PARP	0.00628930817610063
26	VDAC	0.00628930817610063
27	BAD	0.0062111801242236
28	PUMA	0.00617283950617284
29	Noxa	0.00617283950617284
30	DNA Fragmentation	0.00617283950617284
31	APAF1	0.00591715976331361
32	POD	0.00591715976331361
33	ICAD-CAD	0.00588235294117647
34	CAD	0.00558659217877095
35	p53	0.00531914893617021
36	ATM	0.00523560209424084
37	Akt1	0.005
38	p90RSK	0.005
39	FLIP	0.00495049504950495
40	Apo2L/Apo3L-FADD-Procaspase8	0.00495049504950495
41	Fas-FADD-Procaspase8	0.00495049504950495
42	TNFR1-Procaspase8	0.00495049504950495
43	Chk2	0.00469483568075117
44	MDM2	0.00436681222707424
45	p53AIP1	0.00432900432900433

46	PKC	0.00418410041841004
47	PI3K	0.00418410041841004
48	p14(ARF)	0.00367647058823529
49	Growth Factor Receptor	0.00359712230215827

In terms of closeness centrality, IKKs and NF-kB-IkB alpha, etc., are the most crucial nodes in the network. And Growth Factor Receptor and p14(ARF), etc., are the most trivial nodes in the network.

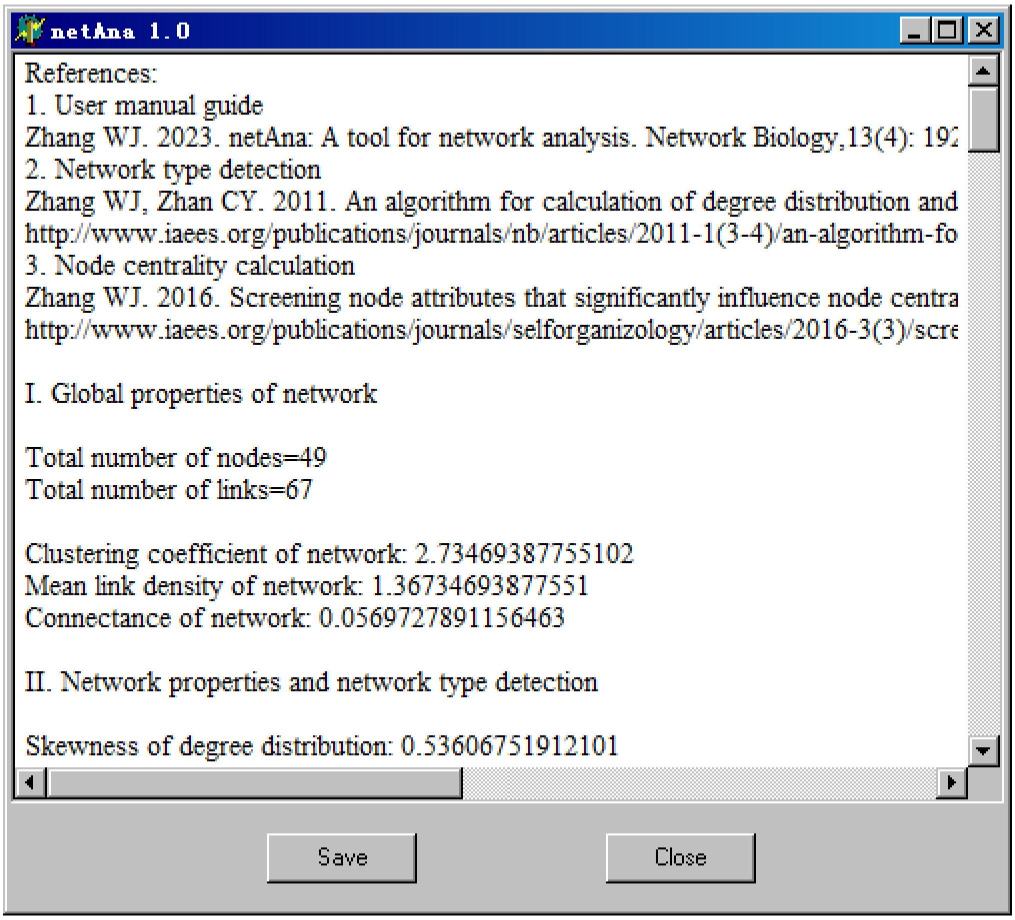
Betweenness centrality

Rank	Node	Betweenness centrality
1	Caspase8	2.94477911646586
2	Apo2L/Apo3L-FADD-Procaspase8	1.93373493975904
3	BID	0.938755020080321
4	p53	0.76004016064257
5	MMP	0.613453815261044
6	IKKs	0.610441767068273
7	Chk2	0.610441767068273
8	BCL2	0.345381526104418
9	TNFR1-TRAF2	0.336345381526104
10	Apoptpsis	0.295180722891566
11	BCLXL	0.295180722891566
12	NF-kB-IkB alpha	0.28714859437751
13	BAD	0.261044176706827
14	Noxa	0.234939759036145
15	Caspase3	0.189759036144578
16	tBID	0.178714859437751
17	BAK	0.141566265060241
18	DNA Fragmentation	0.13855421686747
19	Caspase7	0.134538152610442
20	Akt1	0.133534136546185
21	AIF	0.132530120481928
22	MDM2	0.0953815261044177
23	p90RSK	0.0913654618473896
24	PI3K	0.0893574297188755
25	ICAD-CAD	0.0883534136546185
26	ATM	0.0833333333333333
27	CAD	0.0803212851405622
28	Cytoc	0.0773092369477912
29	HTRA2	0.0742971887550201
30	CIAP1	0.0742971887550201
31	Caspase9	0.073293172690763
32	BIM	0.0592369477911647
33	APAF1	0.0582329317269076
34	BAX	0.0481927710843374

35	VDAC	0.0481927710843374
36	PUMA	0.0481927710843373
37	SMAC	0.0481927710843373
38	Arts	0.0481927710843373
39	EndoG	0.0481927710843373
40	p53AIP1	0.0481927710843373
41	p14(ARF)	0.0481927710843373
42	POD	0.0481927710843373
43	PKC	0.0471887550200803
44	FLIP	0.0471887550200803
45	PARP	0.0471887550200803
46	Growth Factor Receptor	0.0471887550200803
47	TNFR1-Procaspase8	0.0461847389558233
48	Fas-FADD-Procaspase8	0.0441767068273092
49	NF-kB	0.0240963855421687

In terms of betweeness centrality, Caspase8 and Apo2L/Apo3L-FADD-Procaspase8, etc., are the most crucial nodes in the network.

And NF-kB and Fas-FADD-Procaspase8, etc., are the most trivial nodes in the network.



References:
1. User manual guide
Zhang WJ. 2023. netAna: A tool for network analysis. Network Biology,13(4): 192
2. Network type detection
Zhang WJ, Zhan CY. 2011. An algorithm for calculation of degree distribution and
http://www.iaees.org/publications/journals/nb/articles/2011-1(3-4)/an-algorithm-fo
3. Node centrality calculation
Zhang WJ. 2016. Screening node attributes that significantly influence node centra
http://www.iaees.org/publications/journals/selforganizology/articles/2016-3(3)/scre

I. Global properties of network

Total number of nodes=49
Total number of links=67

Clustering coefficient of network: 2.73469387755102
Mean link density of network: 1.36734693877551
Connectance of network: 0.0569727891156463

II. Network properties and network type detection

Skewness of degree distribution: 0.53606751912101

Fig. 3 Results of network analysis from netAna.

References

Chen JL, Zhang WJ. 2020. Parasites govern the topological properties of food webs: a conclusion from network analysis. Applied Ecology and Environmental Research, 18(2): 3419-3437

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

Jiang LQ, Zhang WJ, Li X. 2015. Some topological properties of arthropod food webs in paddy fields of South China. Network Biology, 5(3): 95-112

Jiang LQ, Zhang WJ. 2015. Determination of keystone species in CSM food web: A topological analysis of network structure. Network Biology,5(1): 13-33

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

Xin SH, Zhang WJ. 2020. Construction and analysis of the protein-protein interaction network for the olfactory system of the silkworm *Bombyx mori*. Archives of Insect Biochemistry and Physiology, 105(3): e21737

Xin SH, Zhang WJ. 2021. Construction and analysis of the protein-protein interaction network for the detoxification enzymes of the silkworm, *Bombyx mori*. Archives of Insect Biochemistry and Physiology, 108(4): e21850

Yang S, Zhang WJ. 2022. Systematic analysis of olfactory protein-protein interactions network of fruitfly, *Drosophila melanogaster*. Archives of Insect Biochemistry and Physiology, 110(2): e21882

Zhang GL, Zhang WJ. 2019. Protein-protein interaction network analysis of insecticide resistance molecular mechanism in *Drosophila melanogaster*. Archives of Insect Biochemistry and Physiology, 100(1): e21523

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

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

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

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

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

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

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

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

Zhang WJ. 2018a. Finding maximum flow in the network: A Matlab program and application. Computational Ecology and Software, 8(2): 57-61

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

Zhang WJ. 2021. Construction and analysis of the word network based on the Random Reading Frame (RRF) method. Network Biology, 11(3): 154-193

Zhang WJ. 2023. A desktop calculator for effect sizes: Towards the new statistics. Computational Ecology and Software, 13(4): 136-181

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, Qi YH. 2023. Construction and analysis of acupoint network with functional similarity in Traditional Chinese Medicine. Network Biology, 13(3): 74-83

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