实验二 频繁模式挖掘

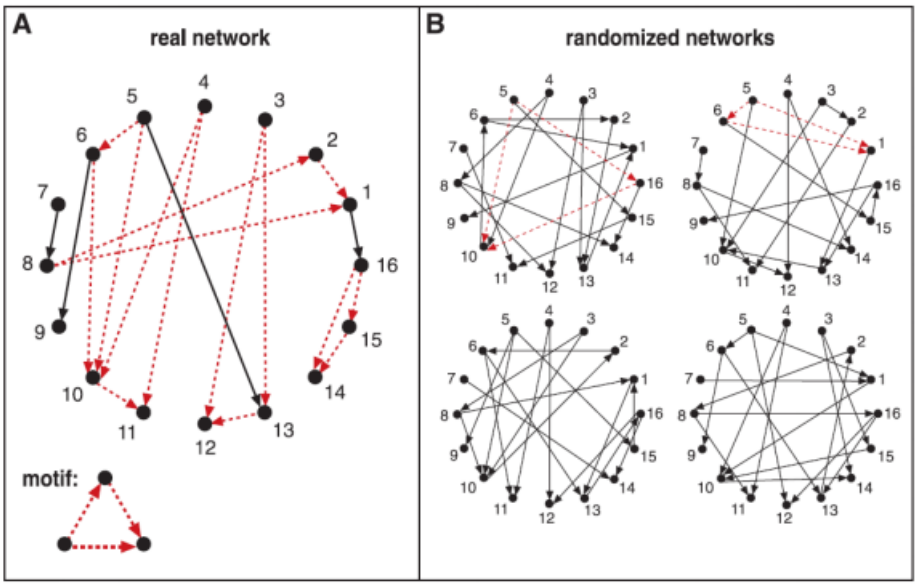
学号：19049100002　　姓名:张泽群　　任课老师：马小科

1. 实验内容

本实验基于现有的频繁子图搜索算法，在确保频繁子图频率精度的条件下提高搜索效率，高效地枚举网络中的所有3阶子图。对网络分析的结果表明，许多网络本质上具有共同的全局性质，如小世界性质和无标度性质。而网络模体则是非常重要的一种网络局部性质。短短几年对网络模体的分析已经取得了大量的研究成果。网络模体是基本的信息处理模块,它们互相配合完成信息处理的工作。然而，模体发现却是一个非常复杂的问题，其中有许多难题需要更深入地研究。

本实验利用蚜虫-巴克纳氏共生体(Buchnera aphidicola)的新陈代谢网络数据集，这是一种研究较为广泛的有向网络。这个新陈代谢网络中的节点代表了生化分子，如蛋白质、脂类、糖、核酸等。网络中的边代表了物质间相互作用的关系，如某一物质经过生化反应生成其他物质。

本实验将数据集转化为邻接矩阵，计算真实网络中13种三元组模体各自出现的个数，然后依据真实网络生成随机网络，并生成100个随机网络，统计这100个随即网络的motif均值，标准差，大于等于和小于等于真实数据集motif的个数，给出Motif数目统计表格，最后实现结果的可视化。



2. 分析及设计

2.1 导入网络数据

我们利用有向图邻接矩阵A来存储网络，0表示无关联，1表示有关联。由于数据集中的节点序号是用字符+编码表示，如'C00016'，实际上节点个数只有270个，因此我们在导入数据的过程中将节点序号转化为1-270的数字

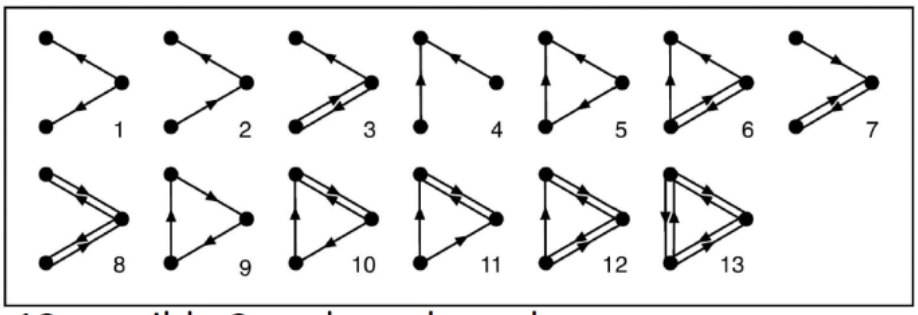
2.2 计算13种三元组模体各自出现的个数

模体（Motif）是指序列中局部的保守区域，或者是一组序列中共有的一小段序列模式。更多的时候是指有可能具有分子功能、结构性质或家族成员相关的任何序列模式。在蛋白质、DNA、RNA序列中都存在

这里我们根据13种网络的特点，对于三元组motif来说，一共存在3个点和最多6条边，因此在判断三元组motif时，只要根据6条边和13种三元组motif的对应关系即可完成判断，并统计其数量。例如第一个motif的关系为

```
（p2->p1）==1 && （p2->p3）==1 && （p1->p2）==0 && （p1->p3）==0 && （p3->p1）==0 && （p3->p2）==0
```

这里我利用3个循环找出所有满足判断条件的点，最后删除重复的点，即可计算获得三元组模体各自出现的个数。



13 possible 3-nodes subgraphs

2.3 产生与真实网络具有相同规模的随机网络

这里我依照所给的步骤生成与真实网络具有相同规模的随机网络。在重复随机交换链接对时，我把新生成的链接对重新放入边集中，从而得到随机化较好的网络。

Step1. 从真实网络开始，重复随机交换连接对。即：$X1 \rightarrow Y1, X2 \rightarrow Y2$替换成$X1 \rightarrow Y2, X2 \rightarrow Y1$ 。

注： 如果$X1 \rightarrow Y2$ 或 $X2 \rightarrow Y1$已经存在，则不进行交换操作（需要重新随机选择连接对)。

Step2. 重复上述过程，直到得到较好的随机化网络。这里我重复指定次数为100次。

为保证与真实网络具有相同规模，我编写函数计算随机网络和真实网络各个节点的度，从而完成验证。

2.4 数据的统计

然后就是简单的数据统计，对每一个随机网络，计算13种三元组模体各自出现的个数，对100个随机网络，计算13种三元组模体各自出现个数的均值、标准差，计算大于等于小于等于真实集的个数。

最后结合这些计算出来的数据，生成一个motif识别结果统计表。

2.5 可视化聚类结果

利用matlab导出所需的聚类结果数据，使用Cytoscape绘制真实网络图和随即网络图。

3. 详细实现

实验使用的编程语言为*matlab*，编程环境为*MATLAB R2021b*。

3.1 数据预处理

由于matlab的load函数无法导入字符串，我们在txt中利用'替换'功能将所有'C'删除。

3.2 导入网络数据并进行数据处理

```
%% 1.导入蚜虫-巴克纳氏共生体的新陈代谢网络并进行数据处理
load('buc_net_s.txt');

pts_set = unique(buc_net_s);   % 获得网络节点集合
pts_number = length(pts_set); % 获得网络节点个数，为270个

% 重新设置节点的序号：1-270
for i=1:pts_number
    buc_net_s(buc_net_s==pts_set(i))=i;
end
```

3.3 产生随机网络和计算13种三元组模体各自出现的个数

函数主体部分：

```
%% 3.产生与真实网络具有相同规模的随机网络100个
%% 4. 对每一个随机网络，计算13种三元组模体各自出现的个数

% 存放100个随机网络13种motif数量的数组
[num1_set,num2_set,num3_set,num4_set,num5_set,num6_set,num7_set,num8_set
,num9_set,num10_set,num11_set,num12_set,num13_set]=deal(zeros(100,1));

for i=1:100
    A_random = random_network(A,buc_net_s,100); % 通过random_network函数生
成随机网络，可以指定重复次数

[num1_set(i),num2_set(i),num3_set(i),num4_set(i),num5_set(i),num6_set(i)
,num7_set(i),num8_set(i),num9_set(i),num10_set(i),num11_set(i),num12_set
(i),num13_set(i)]=find_motif13(A_random);
end
```

生成随机网络主体部分：

```
for i=1:repeat_num % 可重复指定次数
    rand_edge1 = floor(randi(edge_number));   % 随机生成两条边的序号
    rand_edge2 = floor(randi(edge_number));
    X1 = edge_set(rand_edge1,1);              % 两条边对应的起点和终点
    X2 = edge_set(rand_edge2,1);
    Y1 = edge_set(rand_edge1,2);
    Y2 = edge_set(rand_edge2,2);

    if A_random(X1,Y2)==0 && A_random(X2,Y1)==0 && X1~=Y2 && X2~=Y1 % 若
起点和终点不相同且新生成的边不存在
        A_random(X1,Y2)=1;
        A_random(X2,Y1)=1;
        A_random(X1,Y1)=0;
        A_random(X2,Y2)=0;
        edge_set(rand_edge1,:)=[X1,Y2;]; % 将新边放在边集中
        edge_set(rand_edge2,:)=[X2,Y1;];
    end
end
```

计算13种模体主体部分：这里删去重复部分是我发现规律后的改进版本，原始版本详见'test_motif13.m'

```
for i=1:pts_number % 点 1
    for j=1:pts_number % 点 2
        for k=1:pts_number % 点 3

            if i~=k && i~=j && j~=k % 三点均不相等
```

```
                % 6条边状态的13种判断条件
                if A(j,i)==1 && A(j,k)==1 && A(i,j)==0 && A(k,i)==0 &&
A(i,k)==0 && A(k,j)==0
                    num1=num1+1;
                ...
                elseif A(j,i)==1 && A(j,k)==1 && A(i,j)==1 && A(k,i)==1
&& A(i,k)==1 && A(k,j)==1
                    num13=num13+1;
                end
            end
        end
    end
end
num1 = num1/2;    % 删去重复部分
num4 = num4/2;
num5 = num5/1;
num6 = num6/2;
num8 = num8/2;
num9 = num9/3;
num11 =num11/2;
num13 = num13/6;
```

3.4 数据计算和统计

注：这里计算大于等于小于等于真实集的个数的函数cmp_motif_num比较简单，就不多赘述。

```
%% 5. 对100个随机网络，计算13种三元组模体各自出现个数的均值、标准差，计算大于等于小于
等于真实集的个数

% 计算 13种三元组模体各自出现个数的均值
average_3motif =
[sum(num1_set)/100;sum(num2_set)/100;sum(num3_set)/100;sum(num4_set)/100
;sum(num5_set)/100;sum(num6_set)/100;sum(num7_set)/100;sum(num8_set)/100
;sum(num9_set)/100;sum(num10_set)/100;sum(num11_set)/100;sum(num12_set)/
100;sum(num13_set)/100;];

% 计算 13种三元组模体各自出现个数的标准差
std_3motif =
[std(num1_set);std(num2_set);std(num3_set);std(num4_set);std(num5_set);s
td(num6_set);std(num7_set);std(num8_set);std(num9_set);std(num10_set);st
d(num11_set);std(num12_set);std(num13_set);];

% 计算 13种三元组模体各自出现大于等于真实集的个数
large_3motif =
cmp_motif_num(ture_motif_num,num1_set,num2_set,num3_set,num4_set,num5_se
t,num6_set,num7_set,num8_set,num9_set,num10_set,num11_set,num12_set,num1
3_set,0);
```

```
% 计算 13种三元组模体各自出现小于等于真实集的个数
small_3motif =
cmp_motif_num(ture_motif_num,num1_set,num2_set,num3_set,num4_set,num5_se
t,num6_set,num7_set,num8_set,num9_set,num10_set,num11_set,num12_set,num1
3_set,1);
```

3.5 数据导出

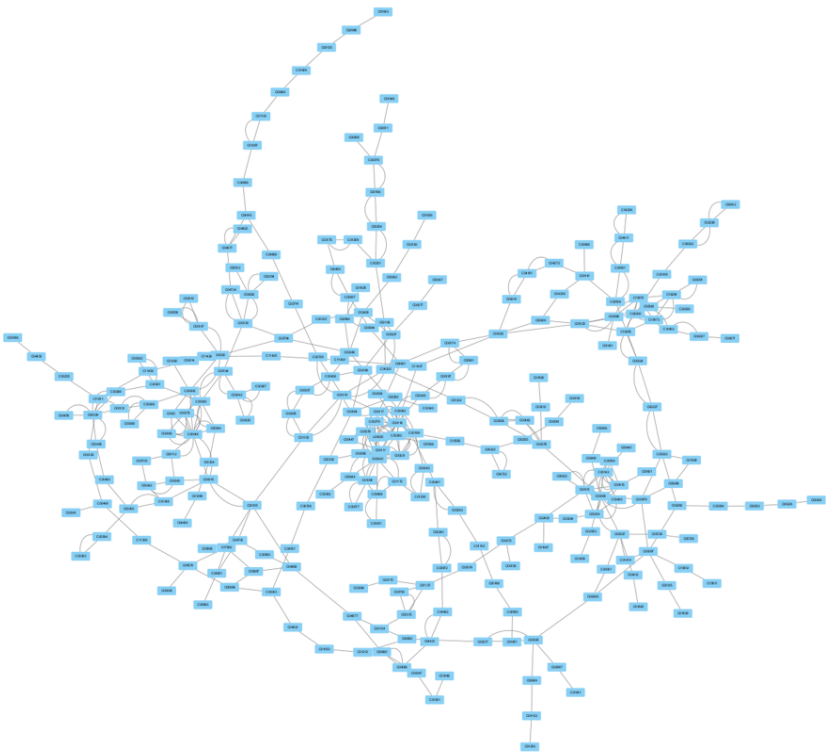
```
%% 6.生成Motif数目统计表格
result_data =
[ture_motif_num,average_3motif,std_3motif,large_3motif,small_3motif;];

xlswrite("motifResult.xls",result_data,'A2:E14');

%% 7.获取画图数据，利用Cytoscape画图
data_edge=[];
for i=1:length(A_random)
    for j=1:length(A_random)
        if A_random(i,j)==1
            data_edge=[data_edge;i,j;];
        end
    end
end
xlswrite("draw_data_edge_100.xls",data_edge,'A2:B443');
```

4. 实验结果

1.Motif识别结果统计表

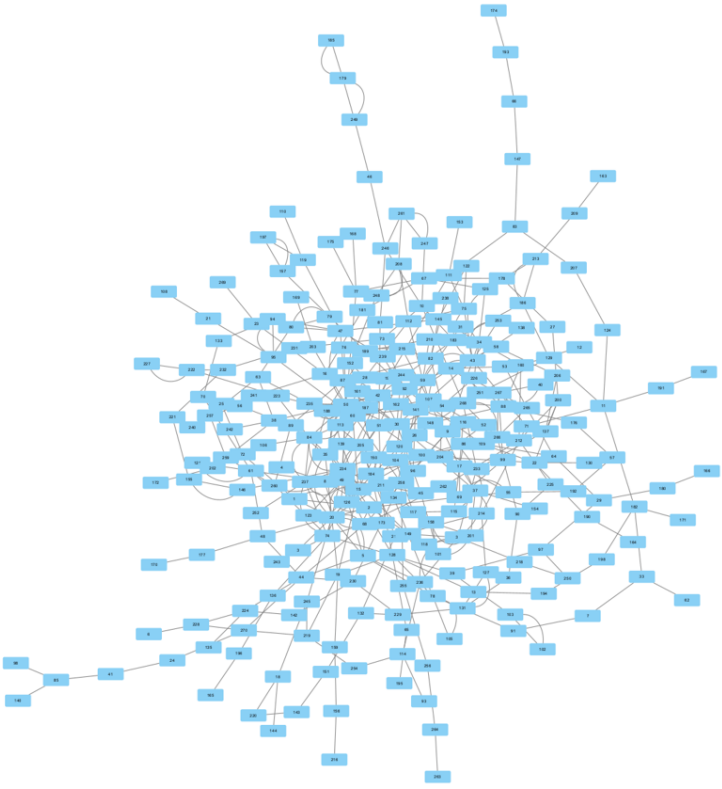
A	B	C	D	E
Motif（真实）	Motif 均值（随机）	Motif 标准差（随机）	≥Motif（真实）的个数	≤Motif（真实）的个数
69	193.52	16.32960997	100	0
143	509.3	36.73148541	100	0
114	137.34	10.76490745	99	2
64	173.91	14.07849209	100	0
1	3.72	2.005447128	98	0
0	1.16	0.992089927	100	29
86	124.43	10.00833491	100	0
154	29.06	8.296062106	0	100
0	1.66	1.273030267	100	22
0	2.59	1.682320427	100	8
1	1.68	1.254325848	85	51
5	3.32	1.656819394	26	92
10	0.71	0.795124029	0	100

2.真实网络图



3.随机网络图

该网络图为100次交换后的随即网络图，在'实验结果图'文件夹中还有200次的。



总结：完成设计方案内容，通过调试，整体完成度较高。但存在的问题是并不知道实验结果的正确性。

5. 心得体会

本次实验的研究学习，我了解到了一种较为简单的频繁模式挖掘算法，对课本上的知识也有了亲身实践，理解更为深入。

对于本次实验也存在一些问题，第一是频繁子图搜索的过程较为笨拙，为枚举网络中的所有3阶子图，只能用于数据较少的情况，第二是我无法判断得出随机网络和motif识别结果统计表的正确性，这是比较苦恼的部分。

6. 源程序与文件附录

源程序：

1. **task2.m**（主函数）
2. **find_motif13.m**（用于计算13种模体的函数，简化版本）
3. **test_motif13.m**（用于计算13种模体的函数，原始版本）。
4. **random_network.m**（用于产生随机网络的函数）
5. **cmp_motif_num.m**（用于计算13种三元组模体各自出现大于等于或小于等于真实集的个数）
6. **or_number.m**（用于计算相似度S，计算两个节点邻居节点的并集）
7. **getDegree.m**（用于计算网络各节点的度，验证网络规模是否相同，主函数中未用到）

其余文件：

1. **buc_net_s.txt**（实验数据）
2. **draw_data_edge_100.xls**（交换次数为100次的，Cytoscape画图数据）
3. **draw_data_edge_200.xls**（交换次数为200次的，Cytoscape画图数据）
4. **motifResult.xls**（Motif数目统计表格）
5. **data.mat**（导入该数据可以获得上述实验效果）
6. 剩余实验结果截图和cys文件均在'实验结果图'文件夹中