数据分析科研营项目报告

2018 年 2 月 13 日

学校：苏州大学

学院：数学科学学院

专业：数学与应用数学

报告人：周楚洋

目录

1 研究背景

不断积累的临床观察表明，生活在人体中的微生物与广泛的人类非传染性疾病密切相关，这为了解复杂的疾病机制提供了有希望的洞见。预测微生物-疾病的相关性不仅可以提高人类疾病的预测和诊断，还可以改善新药的开发。然而，到目前为止，人们还没有试图大规模地了解和预测与人类有关的微生物-疾病相关性。

1.1 miRNA和疾病的关系

MicroRNAs (miRNAs)是一组短链非编码RNA，它们在基因调控中发挥重要作用，miRNAs参与了许多重要的生物过程，包括细胞分化、增殖和凋亡。此外，越来越多的证据表明，miRNAs与各种人类疾病有关。最近已经有许多的经实验验证的miRNA-disease关联。在这些联系的基础上，对各种人类疾病有关的miRNAs进行预测是非常必要的。它有助于为后续的实验研究提供可靠的与已知疾病相关的候选miRNA。

1.2 miRNA、环境因子和疾病的关系

越来越多的证据表明，microRNA(miRNAs)可以与环境因素(EFs)相互作用，从而影响和决定人类疾病。揭示疾病与miRNA - EF相互作用之间的潜在联系，有助于了解miRNA和EF水平的潜在疾病机制，miRNA的特征识别和药物的重新定位。

1.3 lncRNA和疾病的关系

越来越多的证据表明，lncRNA的失调与广泛的人类疾病有关。分析已知的lncRNA-疾病相关性，预测潜在的lncRNA-疾病相关性，并为实验验证提供最可能的lncRNA-疾病对，是必要和可行的。

2 模型介绍

2.1 KATHMDA

2.1.1 模型综述

KATHMAD模型基于相似功能的微生物往往参与类似的疾病关联模式和相似的疾病更可能与功能类似的微生物相关的假设，通过将已知的疾病- 微生物网络、疾病相似网络和微生物相似网络结合起来，在异构网络中整合不同步长的游走，预测了候选微生物-疾病之间的相关性。总的来说，KATHMAD方法是将预测微生物-疾病相关性这个问题转化为图论关系，将微生物和疾病看成图中的结点，综合考虑结点之间的步长，结合KM（微生物的高斯交互形式剖面核相似性）、KD（疾病的高斯交互形式剖面核相似性）和邻接矩阵A来衡量微生物m_i 和疾病d_j之间的相关性。

2.1.2 算法流程

I.由相关的数据库，例如HMDAD、ZINC数据库等，计算邻接矩阵A：

$$A = \begin{cases} 1, & m_i与d_j已知相关; \\ 0, & m_i与d_j关系未知或已知不相关 \end{cases} \tag{1}$$

II.由邻接矩阵A计算$IP(m_i)$与$IP(d_j)$.其中，$IP(m_i)$是来源于A的二进制向量，表示给定微生物m_i与疾病$d_j, (j = 1, 2, ..., n_d)$之间的关系。
III.分别计算微生物和疾病的高斯交互形式剖面核相似性(KM和KD)

$$KM_(m_i, m_j) = exp(-\gamma_m \|IP(m_i) - IP(m_j)\|^2)$$

$$\gamma_m = \frac{\gamma_m^{'}}{\frac{1}{n_m}\sum_{k=1}^{n_m} \|IP(m_k)\|^2}$$

同理，

$$KD_(d_i, d_j) = exp(-\gamma_d \|IP(d_i) - IP(d_j)\|^2)$$

$$\gamma_d = \frac{\gamma_d^{'}}{\frac{1}{n_d}\sum_{k=1}^{n_d} \|IP(d_k)\|^2}$$

IV.为了将KM和KD引入模型的计算中，且使得模型不依赖于微生物-疾病网络的拓扑结构，构建矩阵A^*:

$$A_* = \begin{pmatrix} KM & A \\ A^T & KD \end{pmatrix}$$

V.于是，微生物m_i与疾病d_j存在关系的概率为:

$$S(m_i, d_j) = \sum_{l=1}^{k} \beta^l A^{*l}$$

概率矩阵为：

$$S = \sum_{l \geqslant 1} \beta^l A^{*l} = (I - \beta A^*)^{-1} - I$$

将矩阵S做划分：

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

其中，S_{12}即为微生物与疾病的概率矩阵。

2.1.3 模型评价

KATZHMDA作为一种全球性的计算方法，可以在大规模网络中同时构建所有疾病的潜在微生物-疾病相关性。目前的KATZHMDA模型仍然存在一些限制。首先，由于k(步行数)的最优值可能受到已知微生物-疾病相关性网络的稀疏性的影响，当更多新的微生物-疾病相关性数据被引入数据库时，它的值将需要调整。其次，利用已知的微生物-疾病的相关性计算了高斯交互作用剖面核相似性，从而对那些被研究的疾病和微生物相关性产生偏差。此外，

KATZHMDA的预测性能还不是很理想，额外的数据集成可能会对其增强其预测能力。疾病和微生物的先验信息可以被引入到这个计算模型中，如疾病表型相似性、疾病语义相似性等。最后，KATZHMDA不能应用于没有任何已知关联的新疾病和微生物。引入相似性而不依赖已知的微生物-疾病相关性网络的拓扑信息可以解决这一局限性。

2.2 NRWRH

2.2.1 模型综述

NRWRH是一种基于异构网络的可重启的随机游走方法，该方法将三种不同的网络(蛋白质-蛋白质相似网络、药物-药物相似网络和已知的药物-靶点交互网络)集成到一个异构网络中，由已知的药物-靶点相互作用，实现了在异构网络上的随机游走。

2.2.2 算法流程

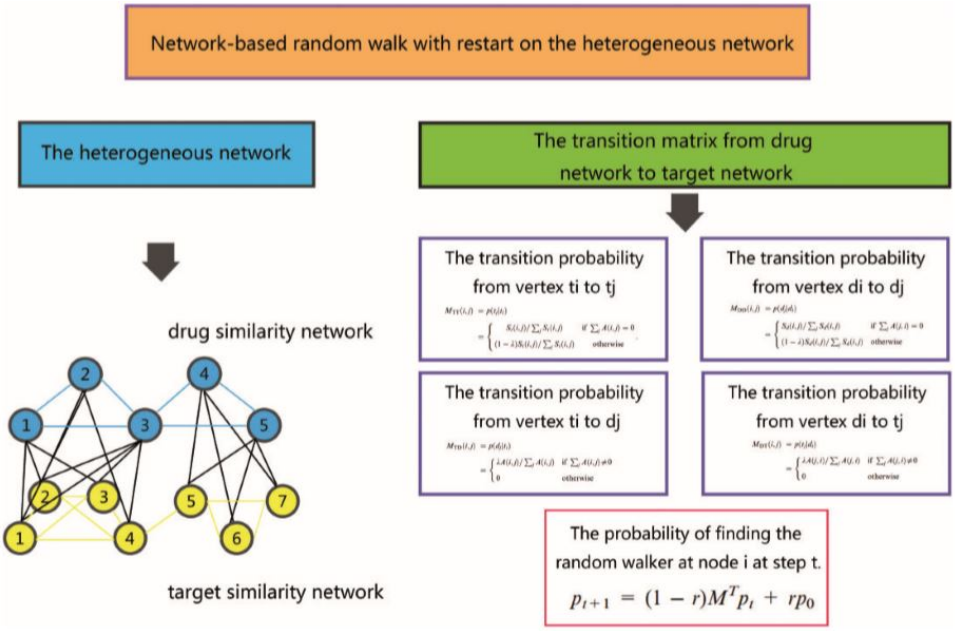


图 1: NRWRH算法流程图.

2.2.3 模型评价

基于相似药物经常针对相似的目标蛋白的假设，以及在随机游走的框架下，该方法可以大规模预测药物-靶点的相关性。与传统的监督或半监督方法相比，NRWRH充分利用网络工具进行数据整合，预测药物-靶点相关性。从异质生物数据中预测潜在的药物-靶点相互作用，不仅有助于了解各种相互作用和生物过程，而且还有助于开发新型药物和改善人类药物。当应用于包括酶、离子通道、GPCRs和核受体等四类重要的药物靶向相互作用时，由交叉验和潜在的药物-靶点相互作用预测方面，NRWRH较以前的方法有了显著的改善。

2.3 IRWRLDA

2.3.1 模型综述

首先，将lncRNA表达相似性、疾病语义相似性和已知的lncRNA-disease关联综合起来，共同确定随机游走的初始概率向量。其次，在基于lncRNA功能相似度和lncRNA高斯交互形式剖面核相似性构建的lncRNA相似网络上实现随机游走。同样的，这种方法也适用于预测miRNA和疾病的相关性上。

2.3.2 算法流程

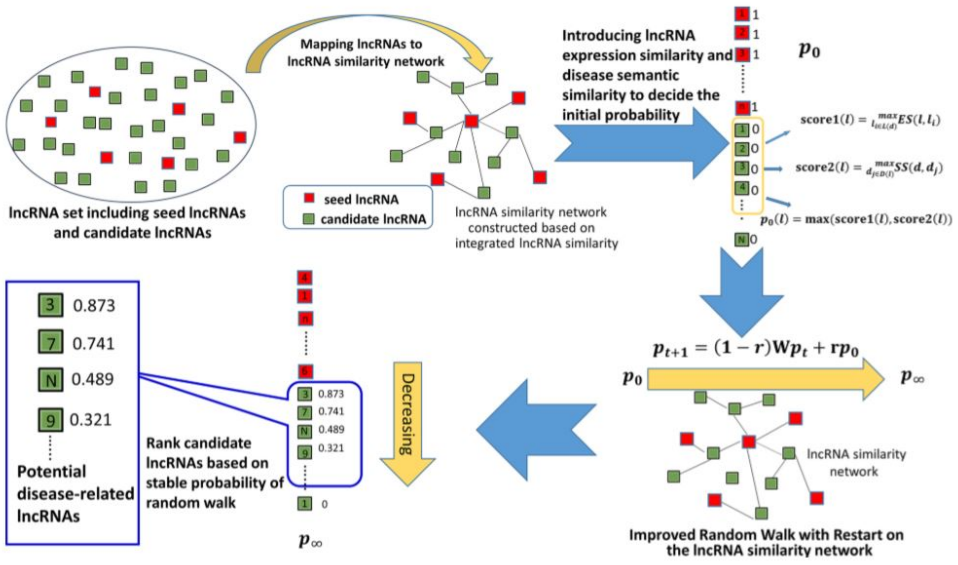


图 2: IRWRLDA算法流程图.

2.3.3 模型评估

从高度可靠的数据库中获得实验证实的lncRNA-疾病（miRNA-疾病）的关联性，并作为训练样本来识别lncRNAs（miRNA）与疾病之间的新型关联。同时，将lncRNA（miRNA）表达相似性、疾病语义相似性和已知的lncRNA-疾病关联（miRNA-疾病关联）综合起来，共同确定随机游走的初始概率向量。这种数据集成可以有效地降低预测偏差。更重要的是，IRWRLDA可以成功地应用于没有任何已知相关lncRNA的新疾病。

2.4 HDMP

2.4.1 模型综述

基于如下假设：1.若有miRNA功能相关，则通常来说与它们有关的疾病在表现型上相似。2.如果miRNA在相同类型的疾病中与某一相似调节模式有关，则它们的靶基因可能具有共同的功能性特征。3.若有miRNA功能相似，则通常来说与它们有关的疾病也相似，反之亦然。因而两种miRNA的功能相似度可以通过两组疾病的语义相似度成功估计得到。为了有效预测疾病miRNAs，我们通过合并疾病的信息内容和疾病之间的表型相似性来计算功能相似

性。此外，miRNA家族或集群的成员被赋予更高的体重，因为他们更可能与类似的疾病相关。由此提出了一种基于k加权最相似邻居（HDMP）的有效预测算法。

2.4.2 算法流程

Step1. 通过合并疾病之间的语义相似性和表现型相似性得到两个miRNA的功能相似性，然后构建miRNA的功能相似性矩阵（利用相似功能的miRNA所关联的疾病往往相似这一特点，通过疾病的相似性得到miRNA的相似性矩阵（对Wang等人算法的改进））
Stpe2.根据miRNA-疾病关系，对每个miRNA家族或簇的成员赋予更大权重。
Step3.考虑miRNA的k加权最相似邻居的功能相似性和在这些邻居中已标记miRNA的分布信息，进而估计每个未标记miRNA的关系得分。
Step4.根据关系得分对所有未标记miRNA进行排名，挑选前几位作为与疾病d相关的候选miRNA。

2.4.3 模型评估

结合了miRNA家族或聚类的权重信息，并考虑特定疾病的分布信息，从而达到有效的预测结果。提出了一种将疾病的信息内容与疾病的表型相似度相结合的测量策略，从而提高了预测两种miRNA功能相似性的准确程度。根据他们与一组疾病的联系，miRNA家族或集群的成员被赋予更高的体重。HDMP结合了与疾病d相关miRNA的功能相似性信息和分布信息来预测miRNA与疾病d相关的可能性，在与现有的一些方法进行比较后，HDMP显得更为可靠。

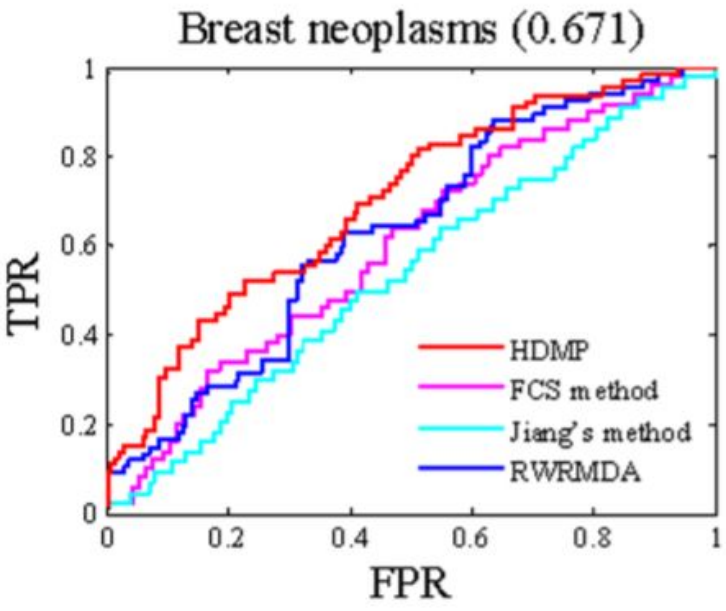


图 3: HDMP与其他算法的比较结果（以乳房瘤为例）.

3 模型验证

3.1 相关名词

真阳性(TP)：预测结果为阳，实际值为阳。
假阳性(FP)：预测结果为阳，实际值为阴。
真阴性(TN)：预测结果为阴，实际值为阴。
假阴性(FN)：预测结果为阴，实际值为阳。
阈值：阈值是人为设定的，在（0，1）中取的值。测试结果高于该阈值的值被认为是阳性，反之为阴性。

3.2 留一交叉验证法

在留一交叉验证的验证框架中，每个已知的微生物-疾病相关性都被排除在测试之外，其他已知相关的微生物-疾病被用作模型的训练样本。具体地说，没有已知相关性证据的所有微生物组将被视为候选样本。进一步得到了相对于候选样本的每一个剩余测试样本的排名。预测等级高于给定阈值的测试样本将被认为是成功预测的。通过设置不同的阈值，我们可以得到相应的真阳性率(TPR，灵敏度)和假阳性率(FPR, 特异性)。在这里，敏感性指的是被预测的比给定阈值高的测试样本的百分比，而特异性指的是被预测的低于给定阈值的测试样本的百分比。通过给定不同的阈值，即阈值遍历（0,1）时，绘制TPR和FPR，得到ROC曲线。进一步计算ROC曲线下去也的面积，即AUC值。AUC值越大，预测结果越好，当AUC值为1时，预测结果较为完美。

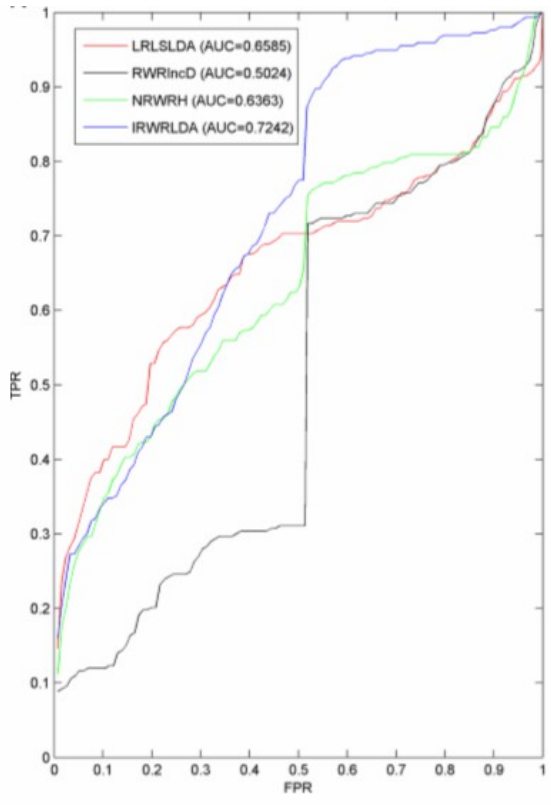


图 4: ROC曲线图例图.

3.3 K折交叉验证法

在k折交叉验证的框架中，所有已知相关的微生物-疾病相关性样本都被随机分为k份。然后选取1份作为模型学习的训练样本，其余部分作为模型评价的测试样本。在类似于留一交叉验证的情况下，所有相关性未知的微生物-疾病对作为候选样本。考虑到随机样本分离对性能评价存在偏差，应将已知的微生物-疾病进行了多次分离，并以与留一交叉验证相似的方式得到相应的ROC曲线和AUC值。

4 算法改进

4.1 基于IRWRLDA方法的算法改进

IRWRLDA方法中，计算初始概率的步骤如下：
对于给定的疾病d，设定已知与d有关的miRNA的初始概率为1，其他的miRNA的初始概率设定为score1和score2中的最大值。取定某个与d关系未知的miRNA m：
Score1:
计算m与已知与d有关的miRNA集合的综合相似性，令M(d)为所有与疾病d已知有关的miRNA集合，则$score1(m) = \max(ES(m, mi),\ mi \in M(d)$.（其中，ES为lncRNA的表性相似性矩阵(expression similarity)）
Score2:
计算疾病d和所有已知与miRNA m 相关的疾病之间的疾病语义相似性，其中D(m)是所有与miRNA m已知有关的疾病的集合，则$score2(m) = \max(SS(d, dj)),\ dj \in D(m)$. 故，miRNA m的初始概率为

$$p0(m) = \max(score1(m), score2(m)).$$

对于初始概率的计算方法可以做一些改进：
若score1(m)只取最大值，不妨设$score1(m) = ES(m, m1)$，则忽略了miRNA m与其他miRNA的相关性，造成了信息的损失，这在已知信息较少的情况下是不合理的。
由此，对score1(m)和score2(m)的算法做如下改进：
令$score1(m) = \frac{sum(ES(m,mj))}{(the\ number\ of\ M(d))},\ mi \in M(d)$;
令$score2(m) = \frac{sum(SS(d,dj))}{(the\ number\ of\ D(m))},\ dj \in D(m)$;
$p0(m) = \frac{score1(m)+score2(m)}{2}$. 这样，初始概率p0(m)综合考虑了miRNA m和已知与疾病d有关的miRNA的信息。通过MATLAB编程实现算法，部分测试结果和MATLAB程序见附录。

4.2 对邻接矩阵A的改进

由于目前已知的疾病与miRNA对较少，在对已知的疾病与miRNA的信息作处理时，应尽量减少信息的丢失。

在一些经典模型中，例如IRWRLDA方法，邻接矩阵A均采用二进制向量来表示。这样忽略了疾病与miRNA相互作用的许多信息，可能会对预测的精度产生影响。以下考虑对邻接矩阵A做一些改进。
由数据库中的数据可以得到已知的与给定疾病d(i) 有关的miRNA。
设所有已知与疾病d(i) 有关的miRNA构成集合S(d(i))={m(i,1), m(i,2) ,⋯, m(i,n)}，调出

与d(i), m(i,1), m(i,2) ,⋯, m(i,n) 相关的实验数据，对$d(i), m(i,1), m(i,2), ,, m(i,n)$根据其生物方面的性质（例如对药物计量的依赖性）进行逐步回归，由回归方程的系数的绝对值判断miRNA与疾病d(i)的相关程度。设回归方程为:

$$y = a_{i1} * m(i,1) + a_{i2} * m(i,2) + ... + a_{in} * m(i,n)$$

令$A(i,j) = |aij|$。这样可以完善数据的信息量，提高预测的准确性。

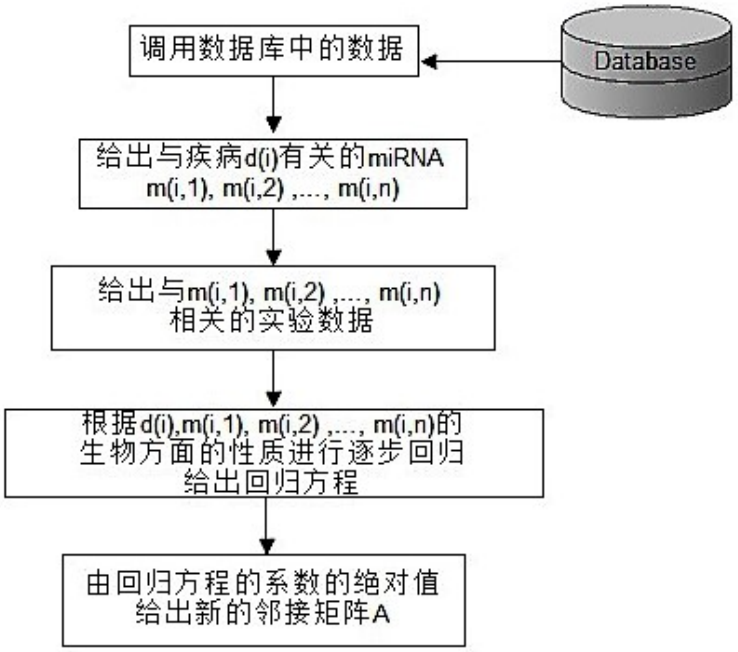


图 5: 对邻接矩阵A的算法改进流程图.

然而，这样的算法也存在一些问题。在对原始数据进行逐步回归时，需要预先对数据进行一些处理，排除异常值对逐步回归的干扰。若原始数据的数据量较大，在对数据处理上可能会花费较多时间。

5 附录

5.1 MATLAB程序

```
clear all;clc;

%导入数据
disease_miRNA_interaction=xlsread('miRNAdisease.xlsx');
SSweighted=textread('疾病语义类似性加权矩阵.txt');
SS=textread('疾病语义类似性矩阵.txt');
FSweighted=textread('miRNA功能类似性加权矩阵.txt');
FS=textread('miRNA功能类似性矩阵.txt');

%计算邻接矩阵A （已知相关为1 否则为0）
for x=1:5430
    i=disease_miRNA_interaction(x,1);
    j=disease_miRNA_interaction(x,2);
    A(i,j)=1;
end
AA=A';

%KD （疾病之间的高斯核剖面相似性矩阵）
%计算gamma_disease
sumnorm2_d=0;
for i=1:383
    norm2_d=norm(AA(i,:),2);
    sumnorm2_d=sumnorm2_d+norm2_d;
end
gamma_disease=1/sumnorm2_d;

%计算KD
for i=1:383
    for j=1:383
        delta_d = norm((AA(i,:)-AA(j,:)),2);
        KD(i,j)=exp(-gamma_disease*delta_d);
    end
end

%KM（miRNA之间的高斯核剖面相似性矩阵）
%计算gamma_miRNA
sumnorm2_m=0;
for i=1:495
    norm2_m=norm(A(i,:),2);
    sumnorm2_m=sumnorm2_m+norm2_m;
end
gamma_miRNA=1/sumnorm2_m;
```

图 6: MATLAB程序1.

```
    %计算KM
for i=1:495
    for j=1:495
        delta_m = norm((A(i,:)-A(j,:)),2);
        KM(i,j)=exp(-gamma_miRNA*delta_m);
    end
end

    %计算疾病的综合相似性矩阵 inte_sim_disease
for i=1:383
    for j=1:383
        if SSweighted(i,j)==1
            inte_sim_disease(i,j)=SS(i,j);
        else
            inte_sim_disease(i,j)=KD(i,j);
        end
    end
end

    %计算miRNA的综合相似性矩阵 inte_sim_miRNA
for i=1:495
    for j=1:495
        if FSweighted(i,j)==1
            inte_sim_miRNA(i,j)=FS(i,j);
        else
            inte_sim_miRNA(i,j)=KM(i,j);
        end
    end
end

    %求初始向量p0
for i=1:383
    t=0;
    xx=[];
    for j=1:495
        if AA(i,j)==1
            t=t+1;      %疾病di有多少个已知相关的miRNA
            xx=[xx,j]; %与疾病di已知相关的miRNA的编号
            score1(i,j)=1;
        else
            score1(i,j)=0;
        end
    end
    for j1=1:495
        if score1(i,j1)==0
            for j2=xx
                score1(i,j1)=score1(i,j1)+inte_sim_miRNA(j1,j2);
            end
            score1(i,j1)=score1(i,j1)/t;
        end
    end
```

图 7: MATLAB程序2.

```
96
97 -  for i=1:495
98 -      t1=0;
99 -      xx1=[];
100 -     for j=1:383
101 -         if A(i,j)==1
102 -             t1=t1+1;      %给定miRNA有多少种已知相关的疾病
103 -             xx1=[xx1,j]; %与给定的miRNA已知相关的疾病的编号
104 -             score2(i,j)=1;
105 -         else
106 -             score2(i,j)=0;
107 -         end
108 -     end
109 -     for j1=1:383
110 -         if score2(i,j1)==0
111 -             for j2=xx1
112 -                 score2(i,j1)=score2(i,j1)+inte_sim_disease(j1,j2);
113 -             end
114 -             score2(i,j1)=score2(i,j1)/t1;
115 -         end
116 -     end
117 - end
118
119     %对两种评分进行综合，得到初始向量p0
120 -   score11=score1';
121 - for i=1:495
122 -     for j=1:383
123 -         score(i,j)=(score11(i,j)+score2(i,j))/2;
124 -     end
125 - end
126
127     %计算W，对miRNA的综合相似性矩阵进行标准化
128 -   D=diag(sum(inte_sim_miRNA,2).^(-1/2)); %生成对角阵
129 -   W=D.*inte_sim_miRNA.*D;
130
131     %进行随机游走
132
133 -   d=input('please input a number between 1 to 383: '); %输入疾病的编号
134 -   p0=score(:,d);
135
136     %设置重启概率r
137 -   r=0.8;
138 -   p1=(1-r)*W*p0+r*p0;
139 -   p2=(1-r)*W*p1+r*p0;
140 - while norm(p1-p2,2)>1e-6
141 -     p1=p2;
142 -     p2=(1-r)*W*p1+r*p0;
143 - end
144
145     %去掉种子结点
146 - for i=1:495
147 -     if A(i,d)==1
148 -         p2(i,1)=0;
149 -     else
150 -         p2(i,1)=p2(i,1);
151 -     end
152 - end
153
154 -   [grade,rank]=sort(p2,'descend');
155 -   result=[rank,grade];
```

图 8: MATLAB程序3.

5.2 MATLAB程序测试

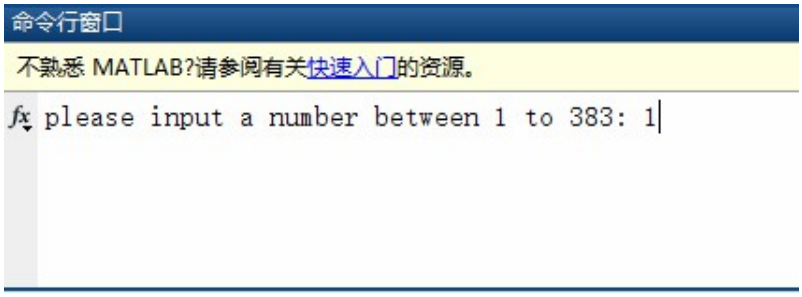
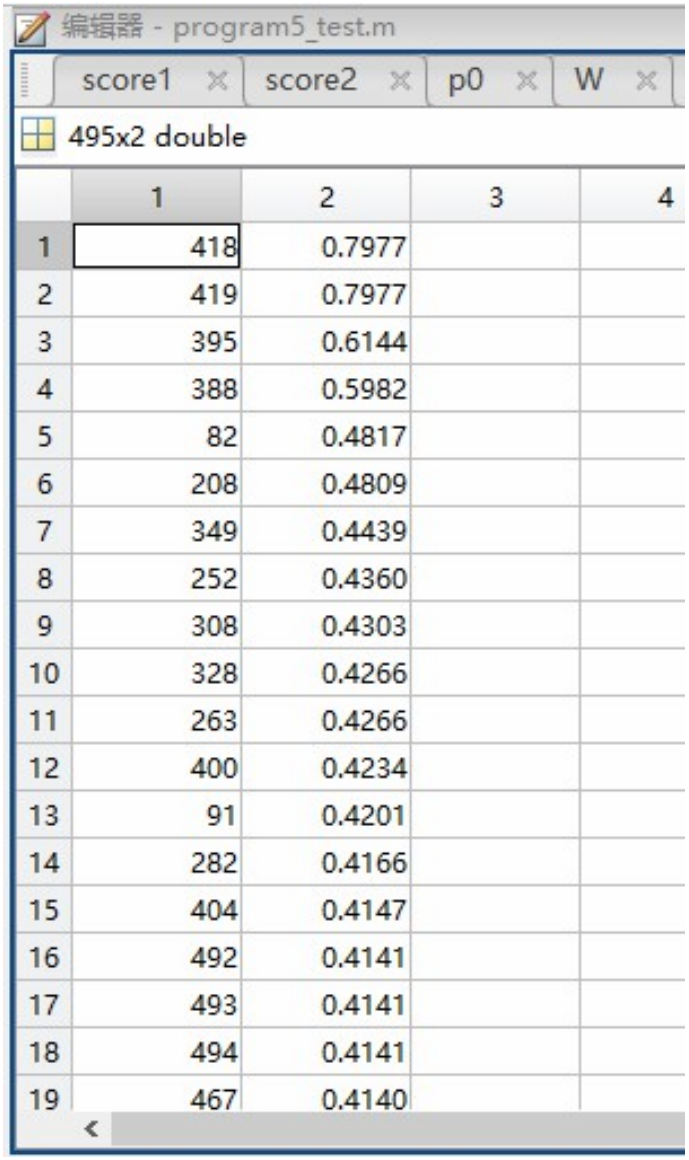


图 9: 测试1.



	1	2	3	4
1	418	0.7977		
2	419	0.7977		
3	395	0.6144		
4	388	0.5982		
5	82	0.4817		
6	208	0.4809		
7	349	0.4439		
8	252	0.4360		
9	308	0.4303		
10	328	0.4266		
11	263	0.4266		
12	400	0.4234		
13	91	0.4201		
14	282	0.4166		
15	404	0.4147		
16	492	0.4141		
17	493	0.4141		
18	494	0.4141		
19	467	0.4140		

图 10: 结果1.

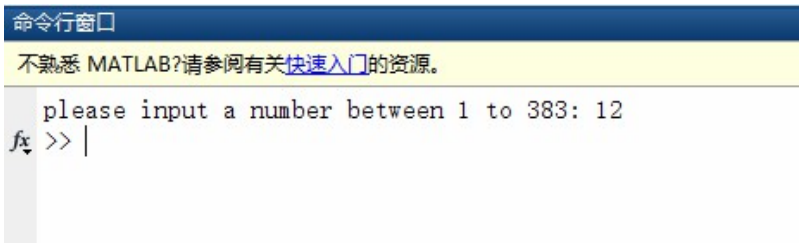


图 11: 测试2.

编辑器 - program5_test.m

rank | AA | grade | result | p

495x2 double

	1	2	3	4
1	404	0.7977		
2	489	0.7976		
3	363	0.7975		
4	396	0.7974		
5	356	0.7974		
6	418	0.7974		
7	419	0.7974		
8	406	0.7973		
9	435	0.7972		
10	424	0.7972		
11	466	0.7971		
12	422	0.7971		
13	473	0.7970		
14	429	0.7970		
15	430	0.7970		
16	170	0.7969		
17	377	0.7969		
18	432	0.7969		
19	476	0.7969		

图 12: 结果2.