Prediction of missing connections in the network: A node-similarity based algorithm

WenJun Zhang
School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 18 August 2015; Accepted 21 September 2015; Published online 1 December 2015

Abstract
In present study, I proposed a node-similarity based algorithm for prediction of missing connections in the network. In this algorithm, whether a node \( v_k \) can connect to \( v_i \) or not, depending on the similarity between \( v_k \) and \( v_i \), the similarities between \( v_i \) and its adjacent nodes, the similarities between \( v_k \) and the adjacent nodes of \( v_i \), and the degree of node \( v_i \), and vice versa. Pearson correlation measure, cosine measure, and (negative) Euclidean distance measure (the three measures are for interval attributes), contingency correlation measure (for nominal attributes), and Jaccard coefficient measure (for binary attributes) were used as the between-node similarity. Two application examples showed a better prediction of the algorithm (approximately 60% of missing connections are successfully predicted). Matlab codes of the algorithm were provided.

Keywords network; connections; prediction; node similarity; algorithm.

1 Introduction
Connection prediction aims to estimate the likelihood of the existence of a connection between two nodes, based on observed connections and the attributes of nodes (Zhou, 2015). Many biological networks, such as food webs, protein–protein interaction networks and metabolic networks, are incomplete networks due to missing connections. 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. Connection prediction can considerably reduce the experimental costs for connection finding. Moreover, connection prediction 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, connection prediction has attracted wide attention. Numerous papers on this topic have been published (Clauset et al., 2008; Guimerà R, Sales-Pardo, 2009; Barzel and Barabási, 2013; Bastiaens et al., 2015; Lü et al., 2015; Zhang, 2015a, 2015b; Zhang and Li, 2015; Zhao et al., 2015; Zhou, 2015).
In present study, I will propose a node-similarity based algorithm for prediction of missing connections in a network. Matlab codes of the algorithm will be presented for further use.

2 Algorithm

Suppose there is an incomplete network, \( X \), with \( m \) nodes (Zhang, 2012a), its adjacency matrix is \( d=(d_{ij})_{m \times m} \). \( d_{ij} = 1 \), if two nodes \( v_i \) and \( v_j \) are adjacent, and \( d_{ij} = 0 \), if \( v_i \) and \( v_j \) are not adjacent; \( i, j = 1, 2, \ldots, m \). Adjacency matrix \( d \) is a symmetric matrix, i.e., \( d=d' \). Known \( n \) attributes for \( m \) nodes. The raw data matrix is \( a=(a_{ij})_{m \times n} \).

Pearson correlation measure, cosine measure, and (negative) Euclidean distance measure (the three measures are for interval attributes), contingency correlation measure (for nominal (1, 2, 3...) attributes), and Jaccard coefficient measure (for binary (0, 1) attributes) can be as the between-node similarity (Zhang, 2016).

Pearson correlation measure is (Zhang, 2011; Zhang et al., 2014; Zhang, 2012a, b; Zhang and Li, 2015)

\[
rij = \frac{\sum_{k=1}^{n} (a_{ik} - a_{ib})(a_{jk} - a_{jb})}{\sqrt{\sum_{k=1}^{n} (a_{ik} - a_{ib})^2 \sum_{k=1}^{n} (a_{jk} - a_{jb})^2}} \\
i, j = 1, 2, \ldots, m
\]

where \(-1 \leq r_{ij} \leq 1\), \( a_{ib} = \sum_{k=1}^{n} a_{ik} / n \), \( a_{jb} = \sum_{k=1}^{n} a_{jk} / n \), \( i, j = 1, 2, \ldots, m \).

Cosine measure is (Zhang, 2007; Zhang, 2012a)

\[
rij = \frac{\sum_{k=1}^{n} a_{ik} a_{jk}}{\sqrt{\sum_{k=1}^{n} a_{ik}^2 \sum_{k=1}^{n} a_{jk}^2}} \\
i, j = 1, 2, \ldots, m
\]

Euclidean distance measure is (Zhang, 2007, 2012a)

\[
d_{ij} = \sqrt{\sum_{k=1}^{n} (a_{ik} - a_{jk})^2}
\]

Thus its negative value is used as the similarity measure

\[
r_{ij} = -d_{ij}
\]

Contingency correlation measure is (Zhang, 2007, 2012a; Zhang et al., 2014):

\[
r_{ij} = 2(h/(s (p-1)))^{1/2} - 1 \quad i, j = 1, 2, \ldots, m
\]

where \(-1 \leq r_{ij} \leq 1\), and

\[
h = s \times (\sum_{i=1}^{p} \sum_{j=1}^{p} s_{ij}^2 / (s_{i.} s_{.j}) - 1) \\
\frac{s = \sum_{i=1}^{p} s_{i.}}{s_{i.} = \sum_{j=1}^{p} s_{ij} \quad n_{.j} = \sum_{i=1}^{p} s_{ij}}
\]

where there are \( p \) available nominal values, i.e., \( t_1, t_2, \ldots, t_p \) for attributes \( i \) and \( j \); \( s_{il} \) is the number of attributes of node \( i \) takes value \( t_k \) and node \( j \) takes value \( t_l \), \( k, l = 1, 2, \ldots, p \).

Jaccard coefficient measure is (Zhang, 2015b)

\[
r_{ij} = (e-(c+b))/(e+c+b) \quad i, j = 1, 2, \ldots, m
\]
where \(-1 \leq r_{ij} \leq 1\), \(c\) is the number of node pairs of 1 for attribute \(i\) but not for \(j\); \(b\) is the number of node pairs of 1 for attribute \(j\) but not for \(i\); \(e\) is the number of node pairs of 1 for both attribute \(i\) and attribute \(j\).

Between-node similarity matrix, \(r=(r_{ij})_{m \times m}\), is a symmetric matrix, i.e., \(r=r'\). In this algorithm, whether a node \(v_k\) can connect to \(v_i\) or not, depending on the similarity between \(v_k\) and \(v_i\), the similarities between \(v_i\) and its adjacent nodes, the similarities between \(v_k\) and the adjacent nodes of \(v_i\), and the degree of node \(v_i\), and vice versa. The procedures of the algorithm for prediction of missing connections are as follows.

1. For each node \(v_i, i=1, 2, \cdots, m\), and \(\forall v_j \in S_i=\{v_k|d_{ik}=1\}\), calculate mean similarity between \(v_i\) and \(\forall v_j \in S_i\), and mean similarity of \(\forall v_j \in S_i\),

\[
i_{\text{mean}}=\text{mean } r_{ij} \quad v_j \in S_i
\]

\[
i_{\text{adj}}_{\text{mean}}=\text{mean } r_{kj} \quad v_k \in S_i, v_j \in S_i
\]

2. For a node \(v_k, i=1, 2, \cdots, m\), the nodes \(\forall v_j \in S_i\), and the node \(v_k \notin S_i, k=1, 2, \cdots, m\). First, calculate the mean similarity of between \(v_k\) and \(\forall v_j \in S_i\)

\[
k_{\text{i_adj}}_{\text{mean}}=\text{mean } r_{ij} \quad v_j \in S_i
\]

then calculate the similarity win

\[
z_{ik}=\alpha(r_{ik} - i_{\text{mean}}) + (1-\alpha)(k_{\text{i_adj}}_{\text{mean}} - i_{\text{adj}}_{\text{mean}})
\]

where \(v_k \neq v_i, d_{ik}=0\), and \(\alpha\) is the importance weight of node \(v_i\) against its adjacent node set \(S_i\) in determining whether the node \(v_k\) can connect to the node \(v_i\) or not, \(0 \leq \alpha \leq 1\). Reverse \(v_k\) and \(v_i\), and repeat the step (1) and (2), calculate \(z_{ki}\).

3. Calculate \(z^{jk}=z_{ki}n_i/(n_k+n_i)+z_{kj}n_j/(n_j+n_k)\), where \(n_i\) is the degree of node \(v_i, i, k=1, 2, \cdots, m; v_k \neq v_i, d_{ik}=0\). The weights, \(n_i/(n_k+n_i)\) and \(n_j/(n_j+n_k)\), are given because the nodes of greater degree are generally more important (Barabasi and Albert, 1999; Zhang and Zhan, 2011; Huang and Zhang, 2012; Zhang, 2012c; Li and Zhang, 2013), and the calculation results on the nodes of greater degree are more statistically confident. Finally, for \(z^{jk} \geq 0\), calculate \(y^{jk}=z^{jk}/2\), to achieve the averaged similarity win, which represents an averaged similarity win of a predicted missing connections against existing connections of the two nodes to be connected.

4. Rank predicted node pairs from the larger \(y^{jk}\) to small ones. The predicted connections with the larger \(y^{jk}\) have higher confidence degree.

5. Once some of the predicted connections are confirmed by observations, the adjacency matrix \(d=(d_{ij})\), can be revised; return step (1) to start new round of prediction.

The following are Matlab codes of the algorithm

```matlab

raw=input('Input the file name of raw data (e.g., raw.txt, raw.xls, etc. The matrix is z=(zij)m×n, where m is total number of nodes, n is the number of attributes): ','s');
adj=input('Input the file name of adjacency matrix or its two-array form (e.g., adj.txt, adj.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; two array form of adjacency matrix, the 1st column is from nodes and 2nd column is to nodes.): ','s');
choice=input('Input a number to choose similarity measure (1: Pearson linear correlation; 2: Cosine measure; 3: (Negative)');
```
alpha=input('Input a weight between 0 and 1 for importance of a node to be connected to against its adjacent nodes (e.g., 0.5, etc. weight=1, means absolute importance of a node and no function of its adjacent nodes): '); 
raw=load(raw); m=size(raw,1); n=size(raw,2); 
adj=load(adj); 
if (size(adj,2)==2) 
nn=size(adj,1); 
adjj=zeros(m); 
for i=1:nn 
adjj(adj(i,1),adj(i,2))=1; 
adjj(adj(i,2),adj(i,1))=1; 
end 
adj=adjj; 
end 
r=zeros(m); 
for i=1:m-1 
for j=i+1:m 
ix=raw(i,:); jx=raw(j,:); 
if (choice==1) 
str='Pearson correlation'; 
ixbar=mean(ix); 
jxbar=mean(jx); 
aa=sum((ix-ixbar).*(jx-jxbar)); 
bb=sum((ix-ixbar).^2); 
cc=sum((jx-jxbar).^2); 
r(i,j)=aa/sqrt(bb*cc); 
end 
if (choice==2) 
str='Cosine measure'; 
aa=sum(ix.*jx); 
bb=sum(ix.^2); 
cc=sum(jx.^2); 
r(i,j)=aa/sqrt(bb*cc); 
end 
if (choice==3) 
str='(Negative) Euclidean distance'; 
r(i,j)=-sqrt(sum((ix-jx).^2)); 
end 
if (choice==4) 
str='Contingency correlation'; 
xx=[ix;jx]; 
pn=1; 
tt(1)=xx(1); 
for kk=1:max(size(xx)) 
jj=0; 
end
for ii=1:pn
    if (xx(kk)==tt(ii)) jj=jj+1; end;
end
if (jj==pn) pn=pn+1; tt(pn)=xx(kk); end;
end
for kk=1:pn
    for jj=1:pn
        temp(kk,jj)=0;
        for ii=1:max(size(ix))
            if ((ix(ii)==tt(kk)) & (jx(ii)==tt(jj))) temp(kk,jj)=temp(kk,jj)+1; end;
        end;
    end;
end;
for kk=1:pn
    pp=0;
    for jj=1:pn pp=pp+temp(kk,jj); end
    ni(kk)=pp;
end
for kk=1:pn
    pp=0;
    for jj=1:pn pp=pp+temp(jj,kk); end
    nj(kk)=pp;
end
summ=0;
for kk=1:pn
    summ=summ+ni(kk);
end;
xsquare=0;
for kk=1:pn
    for jj=1:pn
        if (ni(kk)==0 | nj(jj)==0) continue; end
        xsquare=xsquare+temp(kk,jj)*temp(kk,jj)/(ni(kk)*nj(jj));
    end;
end
xsquare=summ*(xsquare+1);
r(i,j)=2*sqrt(xsquare/(summ*(pn-1)))-1;
end
if (choice==5)
    str='Jaccard coefficient';
    bb=sum((ix==0) & (jx~=0));
    cc=sum((ix~=0) & (jx==0));
    dd=sum((ix~=0) & (jx~=0));
    r(i,j)=(dd-(cc+bb))/(dd+cc+bb);
end
r(j,i)=r(i,j);
end; end
fprintf('
Predicted potential connections, similarity win, and similarity
')
disp('    Node      Node     Similarity win    Similarity')
nn = 0;
res = zeros(m * m, 3);
ilinkmean = zeros(1, m); inlinkmean = zeros(1, m);
num = zeros(1, m); nu = zeros(1, m);
for i = 1:m
    nu(i) = 0;
ilinkmean(i) = 0;
for j = 1:m
    if (i == j) continue; end;
    if (adj(i, j) ~= 0)
u(i) = nu(i) + 1;
    end; end;
    ilinkmean(i) = ilinkmean(i) + r(i, j);
    num(nu(i)) = j;
ilinkmean(i) = ilinkmean(i) / nu(i);
end; end;
ilinkmean(i) = ilinkmean(i) / nu(i);
if (nu(i) > 1)
    for j = 1:nu(i) - 1
        for k = j + 1:nu(i)
            if (k == j) continue; end
            inlinkmean(i) = inlinkmean(i) + r(num(j), num(k));
        end; end;
inlinkmean(i) = inlinkmean(i) / ((nu(i)^2 - nu(i)) / 2);
    end; end;
inlinkmean(i) = r(i, j);
end; end
jinlinkmean = zeros(1, m);
for j = 1:m
    if ((j == i) | (sum(j == num) == 1) | (adj(i, j) ~= 0)) continue; end
jinlinkmean(j) = 0;
    for k = 1:nu(i)
        jinlinkmean(j) = jinlinkmean(j) + r(j, num(k));
    end
jinlinkmean(j) = jinlinkmean(j) / nu(i);
    z = alpha * (r(i, j) - ilinkmean(j)) + (1 - alpha) * (jinlinkmean(j) - inlinkmean(j));
nn = nn + 1; res(nn, 1) = i; res(nn, 2) = j; res(nn, 3) = z;
end; end
ress = zeros(m * m, 4);
mm = 0;
for i = 1:m - 1
    for j = i + 1:m
        if ((res(k, 1) == i) & (res(k, 2) == j)) mm = mm + 1; res(mm, 4) = r(i, j);
            res(mm, 1) = i; res(mm, 2) = j;
            res(mm, 3) = res(mm, 3) + res(k, 3) * nu(i) / (nu(i) + nu(j));
        end; end

if ((res(k,1)==j) & (res(k,2)==i)) ress(mm,3)=ress(mm,3)+res(k,3)*nu(j)/(nu(i)+nu(j)); end
end; end; end
ress(:,3)=round(ress(:,3)/2*10000)/10000;
ress(:,4)=round(ress(:,4)*10000)/10000;
iress=zeros(mm,4);
id=0;
for i=1:mm
if (ress(i,3)>=0) id=id+1; iress(id,:)=ress(i,:); end
end
ires=sortrows(iress(1:id,:),-3);
disp([ires])

Table 1 Predictions of connections between 54 races and populations under different $\alpha$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>47</td>
<td>0.2046</td>
<td>0.5261</td>
<td>19</td>
<td>21</td>
<td>0.1888</td>
<td>0.5132</td>
<td>19</td>
<td>21</td>
<td>0.2352</td>
<td>0.5132</td>
</tr>
<tr>
<td>36</td>
<td>45</td>
<td>0.1969</td>
<td>0.5253</td>
<td>12</td>
<td>21</td>
<td>0.1655</td>
<td>0.4229</td>
<td>12</td>
<td>21</td>
<td>0.1963</td>
<td>0.4229</td>
</tr>
<tr>
<td>23</td>
<td>39</td>
<td>0.1938</td>
<td>0.5064</td>
<td>36</td>
<td>47</td>
<td>0.1608</td>
<td>0.5261</td>
<td>18</td>
<td>21</td>
<td>0.1854</td>
<td>0.3973</td>
</tr>
<tr>
<td>33</td>
<td>47</td>
<td>0.1879</td>
<td>0.5082</td>
<td>33</td>
<td>47</td>
<td>0.1542</td>
<td>0.5082</td>
<td>19</td>
<td>20</td>
<td>0.1766</td>
<td>0.3854</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.1825</td>
<td>0.4208</td>
<td>23</td>
<td>39</td>
<td>0.1502</td>
<td>0.5064</td>
<td>18</td>
<td>20</td>
<td>0.1613</td>
<td>0.3457</td>
</tr>
<tr>
<td>23</td>
<td>32</td>
<td>0.1682</td>
<td>0.4068</td>
<td>36</td>
<td>45</td>
<td>0.1485</td>
<td>0.5253</td>
<td>7</td>
<td>29</td>
<td>0.1601</td>
<td>0.4984</td>
</tr>
<tr>
<td>36</td>
<td>53</td>
<td>0.1584</td>
<td>0.5198</td>
<td>19</td>
<td>20</td>
<td>0.1458</td>
<td>0.3854</td>
<td>5</td>
<td>21</td>
<td>0.15</td>
<td>0.3177</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>0.1576</td>
<td>0.4977</td>
<td>4</td>
<td>8</td>
<td>0.1456</td>
<td>0.4208</td>
<td>11</td>
<td>21</td>
<td>0.1471</td>
<td>0.3115</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>0.149</td>
<td>0.4824</td>
<td>18</td>
<td>21</td>
<td>0.1417</td>
<td>0.3973</td>
<td>5</td>
<td>20</td>
<td>0.1458</td>
<td>0.3088</td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>0.1439</td>
<td>0.2564</td>
<td>14</td>
<td>21</td>
<td>0.139</td>
<td>0.3069</td>
<td>12</td>
<td>20</td>
<td>0.1445</td>
<td>0.3112</td>
</tr>
<tr>
<td>17</td>
<td>26</td>
<td>0.0085</td>
<td>0.1317</td>
<td>3</td>
<td>21</td>
<td>0.0089</td>
<td>0.0669</td>
<td>4</td>
<td>17</td>
<td>0.0086</td>
<td>0.4443</td>
</tr>
<tr>
<td>18</td>
<td>46</td>
<td>0.0051</td>
<td>0.3086</td>
<td>18</td>
<td>47</td>
<td>0.0075</td>
<td>0.3548</td>
<td>12</td>
<td>26</td>
<td>0.0078</td>
<td>0.2327</td>
</tr>
<tr>
<td>15</td>
<td>45</td>
<td>0.0037</td>
<td>0.3155</td>
<td>1</td>
<td>31</td>
<td>0.0072</td>
<td>0.3487</td>
<td>15</td>
<td>48</td>
<td>0.0069</td>
<td>0.3774</td>
</tr>
<tr>
<td>10</td>
<td>29</td>
<td>0.0035</td>
<td>0.3364</td>
<td>7</td>
<td>20</td>
<td>0.007</td>
<td>0.0046</td>
<td>3</td>
<td>6</td>
<td>0.0059</td>
<td>0.0165</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>0.0032</td>
<td>0.4706</td>
<td>10</td>
<td>24</td>
<td>0.0065</td>
<td>0.2771</td>
<td>25</td>
<td>30</td>
<td>0.004</td>
<td>0.3044</td>
</tr>
<tr>
<td>18</td>
<td>54</td>
<td>0.0031</td>
<td>-0.0094</td>
<td>19</td>
<td>24</td>
<td>0.0063</td>
<td>0.2838</td>
<td>37</td>
<td>54</td>
<td>0.0037</td>
<td>0.0746</td>
</tr>
<tr>
<td>1</td>
<td>17</td>
<td>0.0013</td>
<td>0.3344</td>
<td>4</td>
<td>18</td>
<td>0.0058</td>
<td>0.3573</td>
<td>31</td>
<td>54</td>
<td>0.0031</td>
<td>0.069</td>
</tr>
<tr>
<td>11</td>
<td>32</td>
<td>0.0008</td>
<td>0.1182</td>
<td>3</td>
<td>7</td>
<td>0.0043</td>
<td>0.2998</td>
<td>18</td>
<td>35</td>
<td>0.0019</td>
<td>0.387</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>0.0004</td>
<td>0.0246</td>
<td>15</td>
<td>37</td>
<td>0.0034</td>
<td>0.2005</td>
<td>32</td>
<td>42</td>
<td>0.0015</td>
<td>0.4077</td>
</tr>
<tr>
<td>43</td>
<td>54</td>
<td>0.0003</td>
<td>-0.0906</td>
<td>18</td>
<td>37</td>
<td>0.0018</td>
<td>0.2065</td>
<td>4</td>
<td>29</td>
<td>0.0002</td>
<td>0.1191</td>
</tr>
</tbody>
</table>

Node IDs from 1 to 54 represent Lahu-China, Dai-China, Yao-China, Guangdong Han-China, Dulong-China, Buyi-China, Thais, Yi-China, Hunan Han-China, Southern Han-China, Singapore Han-Singapore, Pumi-China, Shanghai Han-China, Liaoning Han-China, Sheggyang Han-China, Northwest Han-China, Northern Han-China, Manchu-China, Japanese, Hokkaido-Japan, Uighur-China, Kazak-China, Siberian Nivkhs population, Siberian Udegeys population, Siberian Koryaks population, Siberian Eskimo, Siberian Chukchi population, South American Indians Ticuna, South American Indians Terena, Siberian Evenki population, Siberian Kets population, USA whites, Spanish, German, Romanians, Bulgarian, Greek, Polish, Turks, Macedonians, Israeli Arabs, Iranian Jews, Ashkenazi Jews-Germany, Libyan Jews, Moroccan Jews, Ethiopian Jews, Native population-Australia’s central desert, Yuendumu Native population-Australia, Kimberley native population-Australia, Cape York native population-Australia, North American blacks, and South African blacks.
3 Application Example

3.1 Analysis of 54 human races and populations and 14 common HLA-DRB1 alleles

Data of the world’s 54 human races and populations (nodes) and 14 common HLA-DRB1 alleles (attributes) (54×14; HLA_DRB1.txt; supplementary material) are from Jia (2001) (Zhang and Qi, 2014). In addition, an adjacency matrix (54×54; HLA_DRB1_adj.txt; supplementary material) and its two array form (×2, HLA_DRB1_adj_twoarrayform.txt; supplementary material) for the network of 54 human races (nodes), derived from linear correlation analysis, is given. In present example, I use Pearson correlation measure. Some results are given in Table 1.

Table 2 Predictions of connections between 12 Chinese populations under different $\alpha$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>9</td>
<td>0.1952 0.2517</td>
<td>7</td>
<td>9</td>
<td>0.2172 0.5798</td>
<td>7</td>
<td>9</td>
<td>0.2577 0.5798</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>0.1878 0.3699</td>
<td>6</td>
<td>12</td>
<td>0.2072 0.58</td>
<td>6</td>
<td>12</td>
<td>0.2379 0.58</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0.184 0.4077</td>
<td>5</td>
<td>9</td>
<td>0.1816 0.41</td>
<td>5</td>
<td>12</td>
<td>0.1795 0.4577</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0.1839 0.41</td>
<td>6</td>
<td>9</td>
<td>0.1812 0.4077</td>
<td>5</td>
<td>9</td>
<td>0.1794 0.41</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>0.1794 0.4577</td>
<td>5</td>
<td>12</td>
<td>0.1795 0.4577</td>
<td>6</td>
<td>9</td>
<td>0.1784 0.4077</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>0.1766 0.5798</td>
<td>4</td>
<td>9</td>
<td>0.1748 0.3699</td>
<td>10</td>
<td>12</td>
<td>0.173 0.4036</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>0.1765 0.58</td>
<td>10</td>
<td>12</td>
<td>0.1629 0.4036</td>
<td>4</td>
<td>9</td>
<td>0.1619 0.3699</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>0.1642 0.345</td>
<td>10</td>
<td>11</td>
<td>0.1536 0.4698</td>
<td>10</td>
<td>11</td>
<td>0.1566 0.4698</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>0.1528 0.4036</td>
<td>3</td>
<td>9</td>
<td>0.1527 0.2517</td>
<td>4</td>
<td>12</td>
<td>0.1461 0.3966</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>0.1506 0.4698</td>
<td>7</td>
<td>12</td>
<td>0.1326 0.2787</td>
<td>7</td>
<td>12</td>
<td>0.1239 0.2787</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>0.1467 0.32</td>
<td>2</td>
<td>9</td>
<td>0.1318 0.345</td>
<td>7</td>
<td>11</td>
<td>0.1198 0.3295</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>0.1413 0.2787</td>
<td>4</td>
<td>12</td>
<td>0.1318 0.3966</td>
<td>8</td>
<td>11</td>
<td>0.1164 0.32</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>0.1384 0.183</td>
<td>8</td>
<td>11</td>
<td>0.1315 0.32</td>
<td>4</td>
<td>11</td>
<td>0.1162 0.4832</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>0.1367 0.3295</td>
<td>7</td>
<td>11</td>
<td>0.1283 0.3295</td>
<td>3</td>
<td>9</td>
<td>0.1101 0.2517</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>0.1302 0.4832</td>
<td>4</td>
<td>11</td>
<td>0.1232 0.4832</td>
<td>2</td>
<td>9</td>
<td>0.0995 0.345</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>0.1176 0.3966</td>
<td>8</td>
<td>12</td>
<td>0.1099 0.183</td>
<td>8</td>
<td>12</td>
<td>0.0813 0.183</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0.1159 0.4808</td>
<td>3</td>
<td>12</td>
<td>0.095 0.2429</td>
<td>3</td>
<td>12</td>
<td>0.0762 0.2429</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>0.1138 0.2429</td>
<td>3</td>
<td>11</td>
<td>0.0754 0.3054</td>
<td>3</td>
<td>11</td>
<td>0.0441 0.3054</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>0.1067 0.3054</td>
<td>2</td>
<td>10</td>
<td>0.0686 0.4808</td>
<td>9</td>
<td>11</td>
<td>0.0402 0.3214</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>0.1046 0.3837</td>
<td>1</td>
<td>9</td>
<td>0.059 0.3837</td>
<td>2</td>
<td>6</td>
<td>0.036 0.5367</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.0695 0.5367</td>
<td>2</td>
<td>6</td>
<td>0.0528 0.5367</td>
<td>9</td>
<td>12</td>
<td>0.0321 0.1284</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.0662 0.3984</td>
<td>9</td>
<td>11</td>
<td>0.0401 0.3214</td>
<td>2</td>
<td>10</td>
<td>0.0214 0.4808</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>0.0457 0.1284</td>
<td>9</td>
<td>12</td>
<td>0.0389 0.1284</td>
<td>1</td>
<td>9</td>
<td>0.0134 0.3837</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>0.04 0.3214</td>
<td>2</td>
<td>5</td>
<td>0.0166 0.3984</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>0.0342 0.004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Node IDs from 1 to 12 represent Tibetan, Uighur, Kazak, Xingjiang Han, Taiwanese, Hong Kong, Northern Han, Shanghai Han, Hunan Han, Manchu, Buyi, and Dai. Successfully predicted missing connections are in bold.

Using a finer adjacency matrix (54×54; HLA_DRB1_adj_Finer.txt; supplementary material) as the “complete network”, it is found that there are 46 missing connections in the incomplete network, HLA_DRB1_adj.txt. Using Pearson correlation, 32 (69.6%), 30 (65.2%), and 30 (65.2%) connections of missing connections for $\alpha=0$, 0.5, 1 respectively, are successfully predicted. These connections are from 177,
161, and 144 predicted connections for $\alpha=0$, 0.5, and 1 respectively. In addition, there are known 255 connections in the incomplete network, HLA_DRB1_adj.txt (potentially maximal 1431 connections). Thus only 12.4% (177/1431), 11.3% (161/1431), and 10.1% (144/1431) of possible connections are needed to be screened for further confirmation respectively, which greatly reduce the cost for experiments and observations.

### 3.2 Analysis of 12 Chinese human populations and 17 HLA-DQB1 alleles

Data of the 12 Chinese human populations (nodes) and 17 common HLA-DQB1 alleles (attributes) (12×17; HLA_DQB1.txt; supplementary material) are from Geng et al. (1995), Chang et al. (1997), Mizuki et al. (1997, 1998), et al. An adjacency matrix (12×12; HLA_DQB1_adj.txt; supplementary material) for the network of 12 human populations, derived from linear correlation analysis, is given. Use Pearson correlation measure and results are given in Table 2.

Using a finer adjacency matrix (12×12; HLA_DQB1_adj_Finer.txt; supplementary material) as the “complete network”, it is found that there are 12 missing connections in the incomplete network, HLA_DQB1_adj.txt. Using Pearson correlation, 7 (58.3%), 7 (58.3%), and 7 (58.3%) connections of missing connections for $\alpha=0$, 0.5, 1 respectively, are successfully predicted. In addition, these connections are from 25, 24, and 23 predicted connections for $\alpha=0$, 0.5, and 1 respectively.

### 4 Discussion

It should be noted that the “complete networks” (with finer adjacency matrices) are defined in a relative sense. More complete networks may exist. Thus the percentage of successfully predicted missing connections may increase with further fining of networks.

In present study, we set $z^{ik}\geq 0$, to calculate $y^{ik} = z^{ik}/2$. However, the threshold can be lowered, for example, $z^{ik} \geq h$, to calculate $y^{ik} = z^{ik}/2$, where $h>0$ is a constant. By doing this, the percentage of successfully predicted connections can be further increased (e.g., 70%, 80%, etc). But at the same time, the percentage of possible connections needed to be screened for further confirmation rises also, which will increase the cost for experiments and observations. Therefore, a compromise between the two percentages is unavoidable and a perfect prediction method is almost impossible.

The present algorithm is based on observed connections and the attributes of nodes. Two extreme situations can be reached by adjusting the $\alpha$. $\alpha=0$ (situation A) means only the similarities between a node ($v_k$) and the adjacent nodes of another node ($v_i$) being prepared to connect to $v_k$ are considered; $\alpha=1$ (situation B) means the comparison of the similarity between a node ($v_k$) and another node ($v_i$) being prepared to connect to $v_k$, and the similarities between $v_i$ and its adjacent nodes, will be made. In practical applications, situation A sometimes occurs. If the mechanism for relationship between node-similarity and connection likelihood is unknown or unsure, $\alpha=0$ is mostly suggested for use. In addition, (negative) Euclidean distance measure can be used in specific cases only, for example, node attributes are spatial coordinates.

The present algorithm is useful to not only the structurally stable networks but also evolving, structurally unstable networks. For structurally stable networks, Lü et al. (2015) proposed a prediction method, structural perturbation method, which was reported to be superior to the known hierarchical structure method (Clauset et al., 2008).

The effectiveness of the present algorithm depends on node attributes and similarity measures. Therefore, future works to improve the present algorithm should mainly focus on (1) selection of the key attributes of nodes in determining connection likelihood, and (2) addition of more specific similarity measures in the algorithm.

Making a little revision on the Matlab codes, the present algorithm can be used to predict which nodes might be connected by a new added node. Further, it can be used to describe network generation and evolution.
Acknowledgment
I am 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
Chang YW, Hawkins BR. 1997. HLA Class I and Class II frequencies of a Hong Kong Chinese population based on bone marrow donor registry data. Human Immunology, 56: 125-135