Article

A node-similarity based algorithm for tree generation and evolution

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 6 August 2015; Accepted 28 September 2015; Published online 1 September 2016

Abstract
In present study we proposed a node-similarity based algorithm for tree generation and evolution. In this algorithm, we assume that each isolated node is a node set at the beginning, two node sets with the greatest similarity tend to connect into a new node set firstly. Repeat this procedure, until all isolated nodes are combined into a tree. Pearson correlation measure, cosine measure, and (negative) Euclidean distance measure (the three measures are for interval attributes), contingency correlation measure (for nominal attributes), or Jaccard coefficient measure (for binary attributes) were used as the between-node similarity. In this way, all connections are sequentially generated and it thus forms the evolution process of a spanning tree of maximum likelihood. The similarity value of a connection can be considered as the weight of the connection. Matlab codes of the algorithm are provided.

Keywords tree; generation; evolution; node similarity; algorithm.

1 Introduction

2 Algorithm
Suppose there are \( m \) isolated nodes (or objects, etc.) and \( n \) attributes. The raw data matrix is \( a=(a_{ij})_{m \times n} \). In the generation/evolution of a tree, I assume that in a set of isolated nodes (node sets), two nodes (node sets) with the greatest similarity tend to connect firstly. 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), or Jaccard coefficient measure (for binary (0, 1) attributes) can be used as the between-node similarity.

Pearson correlation measure is (Zhang, 2011, 2016; Zhang et al., 2014; Zhang, 2012b, c; Zhang and Li, 2015)

\[
    r_{ij} = \frac{\sum_{k=1}^{n} (a_{ik} - \bar{a}_i)(a_{jk} - \bar{a}_j)}{\left(\sum_{k=1}^{n} (a_{ik} - \bar{a}_i)^2 \sum_{k=1}^{n} (a_{jk} - \bar{a}_j)^2\right)^{1/2}} \quad i, j=1, 2, \cdots , m
\]

where \(-1 \leq r_{ij} \leq 1\), \(\bar{a}_i = \frac{\sum_{k=1}^{n} a_{ik}}{n}\), \(\bar{a}_j = \frac{\sum_{k=1}^{n} a_{jk}}{n}\), \(i, j=1, 2, \cdots , m\).

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

\[
    r_{ij} = \frac{\sum_{k=1}^{n} a_{ik} a_{jk}}{\left(\sum_{k=1}^{n} a_{ik}^2 \sum_{k=1}^{n} a_{jk}^2\right)^{1/2}} \quad i, j=1, 2, \cdots , m
\]

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

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

Its negative value is used as the similarity measure

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

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

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

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

\[
    h = s \cdot \left(\sum_{i=1}^{p} \sum_{j=1}^{p} s_{ij} -(s_{ij})^2\right)^{-1} \\
    s = \sum_{i=1}^{p} s_{ik}, \quad s_{i} = \sum_{j=1}^{p} s_{ij}, \quad n_j = \sum_{i=1}^{p} s_{ij}
\]

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

Jaccard coefficient measure is (Zhang, 2015b)

\[
    r_{ij} = \frac{(e-(c+b))}{(e+c+b)} \quad i, j=1, 2, \cdots , m
\]

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., \(r=r^t\).

Calculate the similarity between node sets. Suppose there are two node sets, \(A\) and \(B\). The similarity between \(A\) and \(B\) is defined as the greatest similarity between \(A\) and \(B\)
\[ r_{AB} = \max r_{ij}, \quad i \in A, j \in B \]

At the start, \( m \) isolated nodes (or objects, etc.) are \( m \) node sets respectively. In all of node sets, choose two node sets with the maximal \( r_{AB} \) to combine into a new node set, and the corresponding nodes, \( i \) and \( j \), with the maximal \( r_{ij} \), are connected. Repeat this procedure, until \( m \) isolated nodes are eventually combined into a tree. In this way, all connections are sequentially generated, and thus form the evolution process of a spanning tree (a more general network can be further generated from the spanning tree by adding more connections).

The following are Matlab codes of the algorithm

```matlab
%Reference: Zhang WJ, Li X. 2016. A node-similarity based algorithm for tree generation and evolution. Selforganizology, 3(2): choice=input('Input a number to choose similarity measure (1: Pearson linear correlation; 2: Cosine measure; 3: (Negative) Euclidean distance; 4: Contingency correlation; 5: Jaccard coefficient): '); a=load(str); m=size(a,1); for i=1:m-1 for j=i+1:m ix=a(i,:); jx=a(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; for ii=1:pnn
```
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;
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);
ri.ji=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));
    ri.ji=(dd-(cc+bb))/(dd+cc+bb);
end
r(j,i)=r(i,j);
end; end
adj=zeros(m);
r0=r;
classid=1;
u(classid)=0;
classnum(classid)=m;
for i=1:classnum(classid) x(classid,i)=i; end
tree=zeros(m);
disp(['Step Node Node ' str]);
while (classnum(classid)>1)
    aa=1e+10;
    for i=1:classnum(classid)-1
        for j=i+1:classnum(classid)
            if (r(i,j)>aa) aa=r(i,j); end
        end;
    end;
    aa1=0;
    for i=1:classnum(classid)-1
        for j=i+1:classnum(classid)
            if (abs(r(i,j)-aa)<=1e-06)
                aa1=aa1+1; v(aa1)=i; w(aa1)=j;
                temp=-1e+10;
                for k=1:m
                    if (x(classid,k)==i)
                        for kk=1:m
                            if (x(classid,kk)==j)
                                if (r0(k,kk)>temp) temp=r0(k,kk); end
                            end;
                        end;
                    end;
                end;
                for k=1:m
                    if (x(classid,k)==i)
                        for kk=1:m
                            if (x(classid,kk)==j)
                                if (abs(r0(k,kk)-temp)<1e-06) tree(k,kk)=classid;
                                disp(['num2str(classid) ' 'num2str(k) ' 'num2str(kk) ' 'num2str(r0(k,kk)) ']);
                                adj(k,kk)=1; adj(kk,k)=1;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
    nn1=0;
    for i=1:aa1
        if ((v(i)==0) & (w(i)==0))
            nn1=nn1+1;
        end;
    end;
    for j=1:aa1
        if ((v(j)==v(i)) | (v(j)==w(i)) | (w(j)==w(i)) | (w(j)==v(i)))
            s(v(j))=nn1; s(w(j))=nn1;
        end;
    end;
    v(i)=0; w(i)=0;
end;
end
IAEES
for i=1:nn1
for j=1:classnum(classid)
    if (s(j)==i)
        for k=1:m
            if (x(classid,k)==j) x(classid+1,k)=i; end
        end; end; end
    end; end; end
for i=1:classnum(classid)
    if (s(i)==0)
        nn1=nn1+1;
        for k=1:m
            if (x(classid,k)==i) x(classid+1,k)=nn1; end
        end; end; end;
        classid=classid+1;
        u(classid)=aa;
        classnum(classid)=nn1;
    end; end; end;
for i=1:classnum(classid)-1
for j=i+1:classnum(classid)
    r(i,j)=-1e+10;
    for k=1:m
        if (x(classid,k)==i)
            for kk=1:m
                if (x(classid,kk)==j)
                    if (r0(k,kk)>r(i,j)) r(i,j)=r0(k,kk); end
                end; end; end;
            end; end; end;
        end; end; end;
    r(j,i)=r(i,j);
    end; end; end;
end;
fprintf('Matrix for tree evolution (elements are step IDs)'

'for each step')
tree
fprintf('
')
disp([str ' for each step'])
corr=u(2:classid)
fprintf('
The final adjacency matrix
adj

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) are from Jia (2001) (Zhang and Qi, 2014; HLA_DRB1.txt; supplementary material). Here I use Pearson correlation measure to generate the tree. The process of tree evolution is listed in Table 1.

The distribution of node degrees of final tree for 54 races and populations are indicated in Table 2, and the corresponding tree graph is drawn for convenient comparison using Java software (Zhang, 2012a), as indicated in Fig. 1.
Table 1: The process of tree evolution of world’s 54 human races (nodes) and populations.

<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>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td>28</td>
<td>0.9817</td>
<td>15</td>
<td>16</td>
<td>15</td>
<td>0.9182</td>
<td>29</td>
<td>39</td>
<td>32</td>
<td>0.8791</td>
<td>43</td>
<td>53</td>
<td>54</td>
<td>0.7374</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>35</td>
<td>0.9619</td>
<td>16</td>
<td>20</td>
<td>21</td>
<td>0.9182</td>
<td>30</td>
<td>49</td>
<td>13</td>
<td>0.8753</td>
<td>44</td>
<td>33</td>
<td>37</td>
<td>0.7330</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>26</td>
<td>0.9579</td>
<td>17</td>
<td>45</td>
<td>47</td>
<td>0.9181</td>
<td>31</td>
<td>11</td>
<td>10</td>
<td>0.8750</td>
<td>45</td>
<td>2</td>
<td>7</td>
<td>0.7211</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>11</td>
<td>0.9545</td>
<td>18</td>
<td>16</td>
<td>14</td>
<td>0.9144</td>
<td>32</td>
<td>44</td>
<td>43</td>
<td>0.8731</td>
<td>46</td>
<td>2</td>
<td>11</td>
<td>0.7065</td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>44</td>
<td>0.9527</td>
<td>19</td>
<td>49</td>
<td>50</td>
<td>0.9141</td>
<td>33</td>
<td>10</td>
<td>31</td>
<td>0.8707</td>
<td>47</td>
<td>18</td>
<td>32</td>
<td>0.6789</td>
</tr>
<tr>
<td>6</td>
<td>45</td>
<td>46</td>
<td>0.9458</td>
<td>20</td>
<td>33</td>
<td>39</td>
<td>0.9136</td>
<td>34</td>
<td>46</td>
<td>48</td>
<td>0.8606</td>
<td>48</td>
<td>2</td>
<td>8</td>
<td>0.6581</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>51</td>
<td>0.9413</td>
<td>21</td>
<td>42</td>
<td>40</td>
<td>0.9103</td>
<td>35</td>
<td>5</td>
<td>4</td>
<td>0.8512</td>
<td>49</td>
<td>30</td>
<td>27</td>
<td>0.6562</td>
</tr>
<tr>
<td>8</td>
<td>36</td>
<td>40</td>
<td>0.9371</td>
<td>22</td>
<td>15</td>
<td>18</td>
<td>0.9092</td>
<td>36</td>
<td>18</td>
<td>19</td>
<td>0.8458</td>
<td>50</td>
<td>26</td>
<td>31</td>
<td>0.5708</td>
</tr>
<tr>
<td>9</td>
<td>33</td>
<td>34</td>
<td>0.9333</td>
<td>23</td>
<td>38</td>
<td>43</td>
<td>0.9082</td>
<td>37</td>
<td>5</td>
<td>1</td>
<td>0.8376</td>
<td>51</td>
<td>7</td>
<td>6</td>
<td>0.5526</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>17</td>
<td>0.9328</td>
<td>24</td>
<td>14</td>
<td>12</td>
<td>0.9077</td>
<td>38</td>
<td>1</td>
<td>3</td>
<td>0.8173</td>
<td>52</td>
<td>6</td>
<td>50</td>
<td>0.5353</td>
</tr>
<tr>
<td>11</td>
<td>49</td>
<td>52</td>
<td>0.9259</td>
<td>25</td>
<td>25</td>
<td>27</td>
<td>0.8962</td>
<td>39</td>
<td>3</td>
<td>9</td>
<td>0.8062</td>
<td>53</td>
<td>19</td>
<td>21</td>
<td>0.5132</td>
</tr>
<tr>
<td>12</td>
<td>46</td>
<td>42</td>
<td>0.9252</td>
<td>26</td>
<td>34</td>
<td>22</td>
<td>0.8915</td>
<td>40</td>
<td>22</td>
<td>23</td>
<td>0.8060</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>38</td>
<td>41</td>
<td>0.9199</td>
<td>27</td>
<td>12</td>
<td>11</td>
<td>0.8862</td>
<td>41</td>
<td>33</td>
<td>53</td>
<td>0.7828</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>28</td>
<td>25</td>
<td>0.9185</td>
<td>28</td>
<td>35</td>
<td>40</td>
<td>0.8823</td>
<td>42</td>
<td>29</td>
<td>30</td>
<td>0.7639</td>
<td></td>
<td></td>
<td></td>
<td></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, Sheyang 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, 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.

Table 2: Node degrees for 54 races and populations.

<table>
<thead>
<tr>
<th>ID</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>37</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>46</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>
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. Use Pearson cosine measure and results to generate the tree. The process of tree evolution is as follows.

<table>
<thead>
<tr>
<th>Step</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Cosine measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>8</td>
<td>0.9764</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>10</td>
<td>0.97057</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>4</td>
<td>0.94927</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>5</td>
<td>0.94407</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>7</td>
<td>0.93692</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>3</td>
<td>0.92891</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>12</td>
<td>0.92311</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>2</td>
<td>0.90887</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>1</td>
<td>0.85977</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>11</td>
<td>0.78742</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>9</td>
<td>0.75927</td>
</tr>
</tbody>
</table>

where the node IDs from 1 to 12 represent Tibetan, Uighur, Kazak, Xinjiang Han, Taiwanese, Hong Kong,
Northern Han, Shanghai Han, Hunan Han, Manchu, Buyi, and Dai. The corresponding tree is indicated in Fig. 2.

![Tree Diagram](image)

**Fig. 2** The generated tree of 12 Chinese populations. Node IDs are explained in text.

### 4 Discussion

The present algorithm is based on similarity between nodes. The node sets with greater similarity tend to connect than that with less similarity. The final tree achieved is thus a spanning tree with maximum likelihood. The similarity value of a connection can be considered as the weight of the connection.

Considering the generality of such mechanism of tree generation in nature, it is expected to be a general method for tree generation. This algorithm can produce connections sequentially, thus can reflect the process of tree evolution. Further studies or applications on the algorithm are expected in the future.

### Acknowledgment

We are 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

Cancho RF, Sole RV. 2001. Optimization in complex networks. Santafe Institute, USA
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
Geng L, Imanishi T, Tokunaga K, et al. 1995. Determination of HLA class II alleles by genotyping in a
Manchu population in the northern part of China and its relationship with Han and Japanese populations. Tissue Antigens, 46:111-116


