An Algorithm for Constructing Hypothetical Evolutionary Trees Using Common Mutations Similarity Matrices

Peter Z. Revesz
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588
1 571 201 5639
revesz@cse.unl.edu

ABSTRACT
In this paper, we introduce a new evolutionary tree algorithm that is based on common mutations similarity matrices instead of distance matrices.

Categories and Subject Descriptors
E.1 [Data Structures]: Graphs and Networks.
F.2.2 [Nonnumerical Algorithms and Problems]: Computations on Discrete Structures.

General Terms
Algorithms, Measurement, Performance, Theory.

Keywords
Common mutations similarity matrix, data structure, distance matrix, evolutionary tree, phylogenetic tree.

1. INTRODUCTION
Constructing hypothetical evolutionary trees for a set of genetically related DNA strings, a set of proteins within a protein family, or other related identifiers is a particular clustering or classification problem that occurs frequently in computational biology in various forms [1, 2, 3, 4, 5, 6]. The reconstruction of the evolutionary trees is commonly based on some kind of distance matrix [1].

For example, consider the following seven DNA sequences, $S_1$...$S_7$, each with a length fifteen nucleotides displayed by groups of five nucleotides per column in table below:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>AGCTA</td>
<td>CTAGT</td>
<td>AATCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_2$</td>
<td>AGCTA</td>
<td>CGAGT</td>
<td>AATCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>ATCCA</td>
<td>CTAGT</td>
<td>AACTC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_4$</td>
<td>ATCCA</td>
<td>CTAGT</td>
<td>ATACT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_5$</td>
<td>CGGTA</td>
<td>TTTGT</td>
<td>AAGCT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_6$</td>
<td>CGGTT</td>
<td>CATCA</td>
<td>AATGC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_7$</td>
<td>AGGTA</td>
<td>CCTGA</td>
<td>AATCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Let $S_i[k]$ denote the $k^{th}$ nucleotide of $S_i$. The Hamming distance between two DNA sequences $S_i$ and $S_j$, each with length $n$, denoted $\delta(S_i, S_j)$, is defined as the number of corresponding nucleotide pairs that are different, that is, $\sum_{1 \leq k \leq n} S_i[k] \neq S_j[k]$.

For example, $\delta(S_1, S_2) = 1$ because $S_1[7] = T$ while $S_2[7] = G$. For the above set of DNA sequences the following distance matrix, denoted $D$, can be generated using Hamming distances:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>$S_2$</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>$S_3$</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>$S_4$</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>$S_5$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>$S_6$</td>
<td>9</td>
<td>8</td>
<td>13</td>
<td>13</td>
<td>8</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>$S_7$</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Evolutionary tree construction algorithms generally start from such a matrix to recursive combine pairs of sequences (rows and columns) until only a single combined sequence remains. For example, the UPGMA (unweighted pair group method with arithmetic mean) [6] method would always search for the closest pairs to combine. When several pairs are equally distant, then an arbitrary choice is made. In this case, the closest pairs are $S_1$ and $S_2$ and $S_3$ and $S_4$ because $\delta(S_1, S_2) = 1$ and $\delta(S_3, S_4) = 1$.

Combining the first pair gives:

<table>
<thead>
<tr>
<th></th>
<th>$S_{12}$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{12}$</td>
<td>0</td>
<td>5.5</td>
<td>5.5</td>
<td>6.5</td>
<td>8.5</td>
<td>4.5</td>
</tr>
<tr>
<td>$S_3$</td>
<td>5.5</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>$S_4$</td>
<td>5.5</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>$S_5$</td>
<td>6.5</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>$S_6$</td>
<td>8.5</td>
<td>13</td>
<td>13</td>
<td>8</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>$S_7$</td>
<td>4.5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Then combining the second pair gives:

<table>
<thead>
<tr>
<th></th>
<th>$S_{12}$</th>
<th>$S_{34}$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{12}$</td>
<td>0</td>
<td>5.5</td>
<td>6.5</td>
<td>8.5</td>
<td>4.5</td>
</tr>
<tr>
<td>$S_{34}$</td>
<td>5.5</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>6.5</td>
</tr>
<tr>
<td>$S_5$</td>
<td>6.5</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>$S_6$</td>
<td>8.5</td>
<td>13</td>
<td>8</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>$S_7$</td>
<td>4.5</td>
<td>6.5</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Copyright is held by author/owner(s)
BCB '13, September 22 - 25, 2013, Washington, DC, USA
ACM 978-1-4503-2434-2/13/09.
2. EVOLUTIONARY TREE ALGORITHM

2.1 Common Mutations Similarity Matrix

As a departure from distance matrices, we introduce a common mutations similarity matrix. The motivation behind looking for common mutations is that in practice rare but shared features, such as rare mutations, are often provide useful marker of similarity among a set of closely related items. Moreover, if mutations are rare, then it may be more efficient to count their occurrences than finding the Hamming distances for long sequences. Assuming that the seven DNA sequences are related, we can find the most likely common ancestor sequence, denoted µ, as the mode of each column. If there is no most frequent nucleotide in each column, then we arbitrarily chose one of the most frequent.

Next S12 and S7 would be combined because they are the closest pair of sequences according to the distance matrix. The neighbor-joining method is a more sophisticated and commonly used method that is also based on distance matrices [5].

2.2 A New Evolutionary Tree Algorithm

Our new evolutionary tree algorithm based on common mutations similarity matrices CMSM is described as follows.

ALGORITHM CMSM(S1,…Sn, n)
1 Form n clusters of sequences, each with a single sequence.
2 Find the putative common ancestor µ of the sequences.
3 Construct a graph T with a node for each n cluster and for µ.
4 While (there is more than one cluster)
   5 Find the common mutations similarity matrix.
   6 If (common mutations similarity matrix has non-0 entry)
      7 Merge a closest pair Si and Sj into a new cluster Si
         and create a node for Sij.
      9 Connect the nodes for Si and Sj with parent node Sij.
   10 Else
      11 Connect the remaining clusters’ nodes to parent µ.
   12 Return T.

2.3 Example

Let us consider now the algorithm for the example that we started above. After merging S6 and S7 into a cluster, in the next iteration of the while loop, the algorithm updates the common mutations matrix as follows:

According to the common similarity matrix the closest pair of sequences is S6 and S7. Hence these will be merged. When we merge two sequences Si and Sj in the merged sequence the kth element will be equal to the nucleotide in the two sequences if Si[k] = Sj[k] and will be equal µ[k] otherwise. Hence the matrix of sequences will be updated as follows:

The updated common mutations matrix will be:
Next merging the closest pair, $S_{5}$ and $S_{67}$, yields:

\[
\begin{array}{c}
S_{1} \\
S_{2} \\
S_{34} \\
S_{5} \\
S_{67} \\
\end{array}
\begin{array}{cccc}
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 2 & 0 \\
\end{array}
\]

Next merging the closest pair, $S_{1}$ and $S_{2}$, yields:

\[
\begin{array}{c}
S_{12} \\
S_{34} \\
S_{567} \\
\end{array}
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

Finally, these can be interpreted as being separate branches all descend from the common ancestor sequence $\mu$. Hence in the else clause, the new algorithm connects the remaining clusters’s nodes to $\mu$ and generates the following hypothetical evolutionary tree:

\[
\begin{array}{c}
S_{12} \\
S_{34} \\
S_{567} \\
\end{array}
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

Note that this tree is different from the tree that would be generated by the UPGMA algorithm.

### 2.4 Computational Complexity

The computational complexity of the algorithm can be kept low if we set up a data structure that keeps a list of all mutations for each cluster. Then when merging clusters, then we need to compute only the intersection of the lists associated with the two child clusters.

### 3. ACKNOWLEDGMENTS

Peter Z. Revesz is currently on leave from the University of Nebraska-Lincoln and is an AAAS Science & Technology Policy Fellow and serves as a Program Manager in the U.S. Air Force Office of Scientific Research (AFOSR), a basic research funding agency of the federal government. The current work was not supported by AFOSR, and the views and opinions expressed in this publication are those of the author and do not necessarily reflect the official policy or position of the U.S. government.

### 4. REFERENCES


