Mathematical Understanding of Sequence Alignment and Phylogenetic Algorithms: A Comprehensive Review of Methods

Rashid Saif 1,2*, Sadia Nadeem1, Ali Iftekhar, Alishba Khaliq1, Saeeda Zia3

1 Institute of Biotechnology, Gulab Devi Educational Complex, Lahore, Pakistan
2 Decode Genomics, 323-D, Punjab University Employees Housing Scheme (II), Lahore, Pakistan
3 Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Lahore, Pakistan

*Corresponding author: rashid.saif37@gmail.com

Abstract

Context: Pairwise sequence alignment is one of the ways to arrange two biological sequences to identify regions of resemblance that may suggest the functional, structural, and/or evolutionary relationship (proteins or nucleic acids) between the sequences. There are two strategies in pairwise sequence alignment: Local sequence Alignment (Smith-waterman algorithm) and Global sequence Alignment (Needleman-Wunsch algorithm). In local sequence alignment, two sequences that may or may not be related are aligned to find regions of local similarities in large sequences whereas in global sequence alignment, two sequences same in length are aligned to identify conserved regions. Similarities and divergence between biological sequences identified by sequence alignment also have to be rationalized and visualized in the sense of phylogenetic trees. The phylogenetic tree construction methods are divided into distance-based and character-based methods.

Evidence Acquisition: In this article, different algorithms of sequence alignment and phylogenetic tree construction were studied with examples and compared to establish the best among them to look into background of these methods for the better understanding of computational phylogenetics.

Conclusions: Pairwise sequence alignment is a very important part of bioinformatics to compare biological sequences to find similarities among them. The alignment data is visualized through phylogenetic tree diagram that shows evolutionary history among organisms. Phylogenetic tree is constructed through various methods some are easier but does not provide accurate evolutionary data whereas others provide accurate evolutionary distance among organism but are computationally exhaustive.

Keywords: Sequence alignment methods, Phylogenetic methods, Local sequence alignment, Global sequence alignment, UPGMA, Neighbour joining, Fitch Margoliash, Maximum Parsimony, Maximum Likelihood
Sequence comparison lies at the heart of bioinformatics analysis. As newly biological sequences are generated at exponential rates, sequence comparison is becoming increasingly important. It is a vital step toward structural, functional and evolutionary analysis of the newly determined sequence. The most fundamental method of comparison is sequence alignment. This is the process by which sequences are compared by searching for common character patterns and creating residue-residue correspondence among related sequences. Pairwise sequence alignment is the process of aligning two sequences. Then after aligning all sequences, we move towards the next step that is phylogenetic tree construction methods, to find the evolutionary distance between different sequences [1].

The building blocks of macromolecules nucleotides and amino acids can be considered molecular fossils that tell us about the history of millions of years of evolution. During the time period, sequences accumulate mutations; they undergo random changes and diverge over time. Traces of evolution remain in certain portions of sequences which allow the identification of common ancestry. By comparing sequences through alignment, patterns of conservation and variation can be identified, which reveals the structural, functional and evolutionary relationships between organisms. Phylogenetic trees also tell us about the evolutionary history of organisms. The branching patterns of tree show divergence and convergence of different sequences. For this purpose, we use fossil records which contain information about ancestors of current species and timeline of divergence.

There are two alignment algorithms for pairwise sequence alignment, global sequence alignment and local sequence alignment [2,3]. Both algorithms are based on three methods; dynamic programming method, dot matrix method and word method. Dynamic programming determines optimal alignment by matching possible pairs of characters between the two sequences. The dot-matrix method is a graphical way of comparing two sequences in a two-dimensional matrix. Word data is used in fast database similarity searching. The dynamic programming method is discussed herein. Performing optimal alignment between sequences often involves applying gaps that represent insertions and deletions. Assigning gaps may be less or more arbitrary because there is no evolutionary theory to determine the precise cost for introducing insertion and deletions. For the construction of phylogenetic tree, there are two main categories of tree-building methods one is based on distance and other on discrete characters. The distance method constructs a tree for all taxa based on pairwise distance scores in the matrix. The character-based method or discrete method does not use pairwise distances and it is directly based on sequence characters. Both methods are subdivided into further categories. Distance-based algorithms are subdivided into Unweighted Pair Group Method Using Arithmetic Average (UPGMA), Neighbour-Joining (NJ), Fitch-Margoliash (FM), Minimum-Evolution (ME) and the character-based methods are subdivided into Maximum-Parsimony (MP) and Maximum-Likelihood (ML). All these algorithms construct a phylogenetic tree on the basis of evolutionary changes between different sequences [1,4].
Sequence alignment methods

Alignment strategies

The overall goal of sequence alignment is to find the best pairing of two sequences. By comparing sequences through alignment, patterns of conservation and variation can be identified. There are two main strategies that are often used: Global sequence alignment and the Local sequence alignment.

Global sequence alignment

Two sequences to be aligned are considered to be generally similar over their entire length in the global sequence alignment. Global sequence alignment is carried out between the entire length of two sequences to find the regions of the best possible alignment. This method of alignment is optimal for two closely related sequences of the same length. For the sequences that are divergent or not similar in length, global sequence alignment doesn’t produce optimal results because it fails to recognize the local regions of high similarity between two sequences [1,5].

The Needleman-Wunsch algorithm is the classical global pairwise alignment algorithm employing dynamic programming [2]. Let’s suppose we have two sequences: Sequence 1: ACGTGCTTCA and Sequence 2: CATCCTTG. The global alignment between these two sequences involves the following steps; Initialization, Matrix Filling, Trace backing, Alignment. An example of a global sequence alignment between the entire length of these two sequences is discussed further.

The first step in global sequence alignment is initialization which involves the construction of a square matrix to house both sequences in rows and columns. There are some important rules to consider before proceeding to matrix filling. At the starting of the matrix extra row and column is given for adding gaps. Once the scoring matrix is created the gap, match and mismatch scores are arbitrarily assigned by the user, and they do not have to be specific. The first column and row starting from 0 are filled by adding the gap scores in each previous block (Figure 1). The alignment scoring starts by adding the scores from left, bottom and diagonal box and choosing the highest value to put in that box. If the value is coming from the left box the gap score is added, if the value is coming from the bottom box, the gap score is added, but if the value is coming from the diagonal box, we add match or mismatch score depending on our sequence.
For this example, the scoring scheme is; Gap score: -2, Match score: +5 and Mis-match score: -1. For filling the matrix, the scores from left, bottom and diagonal blocks are considered. As mentioned previously, if the scores are coming from left or bottom blocks the gap score is added into the scores of left and bottom blocks. If the score is coming from the diagonal block, the match or mismatch score is added. If two nucleotide sequences match, the match score is added into the score of the diagonal block and vice versa (Figure 2).

The two sequences are scored at their entire length and the block from which the highest score is obtained is marked to ease trace backing, which is the third step in the global sequence alignment algorithm. When all the cells are filled with scores, the best alignment is determined through a trace-back procedure to search for the path with the best total score. For backtracing,
start from the top box of the matrix towards the origin of the matrix. When a path moves horizontally or vertically, a penalty is applied. The best matching path is the one with the maximum score (Figure 3).

**Figure 3: Global sequence alignment backtracing and sequence alignment**

### Local sequence alignment

The divergence level between the two sequences to be aligned is not easily recognized in the regular sequence alignment. The sequence length of two sequences can sometimes be unequal, and in such case, it is preferred to find the regions of local similarity than to scan to the whole length of two sequences. The Smith-Waterman algorithm is the first algorithm to employ dynamic programming for local sequence alignment [3]. In this algorithm, positive scores are given to the matches and zeros to the mismatches. No negative scores are used in this algorithm. Local sequence alignment is suitable to align divergent sequences at the cost of full-length alignment but it finds the best local regions of similarity between two sequences of different origin [1,6].

The local sequence alignment involves the same overall steps as the global sequence alignment. Initialization, matrix filling, trace backing and alignment. Same two above sequences are used in the following example of local alignment as the global sequence alignment and the scoring scheme used in this example of local sequence alignment is: Gap score: -2, Match score: +1 and Mismatch score: -1.

The matrix initialization begins with assigning both sequences in rows and columns with a gap before each just like global sequence alignment. Gap scores are added to fill the first row and column of the square matrix since negative scores are not allowed in local alignment algorithm; the first row and column is just zeros (Figure 4).
The next step in local sequence alignment is matrix filling. The rules of matrix filling are the same as the global sequence alignment, i.e. if the scores are coming from left or bottom blocks, the gap score is added into the scores of left and bottom blocks. If the score is coming from the diagonal block, the match or mismatch score is added. If two nucleotide sequences match, the match score is added into the score of the diagonal block and vice versa. The only difference is the negative scores cannot be added into the matrix (Figure 5).

After filling the matrix, the next step in local sequence alignment is trace backing. The matrix is traced back from the top to the origin of the matrix to find regions of best scores. The best matching path is the one that has the maximum total score. If two or more paths reach the same highest score, then one is chosen arbitrarily to represents the best alignment. The path can also move horizontally or vertically at a certain point (in case, if the highest score of cells comes from left or bottom cell’s score), which corresponds to the introduction of gap or an insertion or deletion for one of the two sequences. In local alignment, we only trace back local (fully aligned
region without mismatch or gaps) aligned region. The traced back regions are aligned in the last step (Figure 6).

**Figure 6: Local sequence alignment backtracing and sequence alignment**

**Phylogenetic tree construction methods**

Phylogenetics is the study of the evolutionary relationship between organisms using a tree like diagram. The tree branching pattern represents evolutionary relationships between organisms. There are two main methods for phylogenetic tree construction, Distance-Based Methods and Character-Based Methods [7,8]. These methods are further sub-divided into different categories.

**Figure 7: Phylogenetic tree construction methods**
Distance-Based Methods

Phylogenetic trees can be constructed using the distance-based method, which relies explicitly on the genetic distances between sequences. This method requires the sequence data to be transformed into a pairwise similarity matrix for use during tree inference. The distance-based algorithm can be further sub-divided into the clustering-type algorithms and the optimality-based algorithms. The clustering-type algorithms construct trees starting from most similar sequence pairs. They include the unweighted pair group method using arithmetic average (UPGMA) and neighbour-joining (NJ). The optimality-based algorithms compare different tree topologies and select the one that best fits the actual evolutionary distance [1,9-12]. These include Fitch-Margoliash method and the Minimum Evolution method.

Unweighted Pair Group Method Using Arithmetic Average (UPGMA)

Unweighted Pair Group Method Using Arithmetic Average method constructs a rooted phylogenetic tree (dendrogram) using distance matrix obtained from the alignment of different sequences. UPGMA assume that all taxa evolve at a constant rate, however real data rarely meet this assumption [10].

Starting from a distance matrix, two taxa with the smallest pairwise distance are paired. The new pair is treated as a single taxon and the distance between the new taxon and all remaining taxa is calculated to form a reduced matrix. The process is repeated until no further taxa are left. An example of phylogenetic tree construction using the UPGMA algorithm is discussed below.

Consider the four different sequences; Sequence A: AACTGGCTTA, Sequence B: AACTCGGT TA. Sequence C: CATTGGCATA and Sequence D: CCCTGAAACC. Align these sequences to find the mismatched regions (Figure 8).

<table>
<thead>
<tr>
<th>Comparing “A” with other sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A: AACTGGCTTA</td>
</tr>
<tr>
<td>B: AACTGGCTTA</td>
</tr>
<tr>
<td>2. A: AACTGGCTTA</td>
</tr>
<tr>
<td>C: CATTGGCATA</td>
</tr>
<tr>
<td>3. A: AACTGGCTTA</td>
</tr>
<tr>
<td>D: CCCTGAAACC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparing “B” with other sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. B: AACTGGCTTA</td>
</tr>
<tr>
<td>C: CATTGGCATA</td>
</tr>
<tr>
<td>2. B: AACTGGCTTA</td>
</tr>
<tr>
<td>D: CCCTGAAACC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparing “C” with other sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. C: CATTGGCATA</td>
</tr>
<tr>
<td>D: CCCTGAAACC</td>
</tr>
</tbody>
</table>

Figure 8: Multiple sequence alignment (MSA) for UPGMA pairwise matrix construction
A distance matrix is constructed using the MSA data; by using this distance matrix, the UPGMA algorithm starts grouping two taxa with the shortest distance. After grouping, those taxa will consider as a single taxon. A node is placed at the midpoint of those taxa (Figure 9). The distance between this new composite taxon and all next taxa is again calculated, and then the taxon which is at the shortest distance from composite taxon is grouped with it. The same grouping process is repeated and another newly reduced matrix is created. The process continues until all taxa are placed on the tree. Outgroup of tree is the last taxon that is added in the end of the process, producing a rooted tree. In this method, all taxa evolve at a constant rate and they are equally distant from the root (real data rarely meet this assumption). UPGMA often produces erroneous tree topologies. However, owing to its fast speed of calculation, it has found extensive usage in clustering analysis of DNA sequence data.

![Distance Matrix](image)

**Figure 9: UPGMA tree construction from pairwise sequence alignment matrix**

As taxon A and B are at the shortest distance, A-B is combined and treated as a single taxon. In the UPGMA method, all taxa are at equidistant from the node thus branch length of A and B from the node is given as AB/2 = 2/2 = 1; as a result, new composite taxon A-B is created (Figure 9). After that new distances are calculated to construct a reduced matrix. The total distance of C from A and C from B are summed up and divided by 2 to get the new distance of C from A-B. Similarly, the distance of D from A-B is calculated. The taxon with the smallest distance is picked again from this newly constructed distance matrix and added to the tree (Figure 10).
Figure 10: Taxon are combined into a reduced matrix and the tree is extended using new matrix

As taxon C is equidistant from taxon AB (AB-C/2 = 4/2 = 2). The distance of D is calculated from A-B-C and since D is at equidistance from taxon A-B-C (ABC-D/2 = 7/2 = 3.5) and the final outgroup is placed (Figure 11).

Figure 11: Final tree constructed by the UPGMA method, all branches are at equidistance

**Neighbour-Joining Method**

Neighbour-joining is a standard way of constructing a phylogenetic tree. It is somewhat similar to the UPGMA method in that it builds tree step-wise starting from the shortest distance and moving towards the longest distance step by step. NJ method also uses a reduced matrix for tree construction. However, this method does not assume all the taxa to be equidistant from the root; it calculates actual evolutionary rates or distance between sequences [1,11]. This method uses the conversion step to calculate the actual distance of taxa from their root, using the following formula:
\[ d'_{AB} = d_{AB} - \frac{1}{2} (r_A + r_B) \quad \text{(Equ. i)} \]

Whereas \( d'_{AB} \) is the converted distance between A and B, \( d_{AB} \) is the actual evolutionary distance between A and B and \( r_A \) (or \( r_B \)) is the sum of distances of A from all other taxa. The \( r \)-value (for example, \( r_A \)) can be calculated using equation ii. The \( r \)-values are used to construct a reduced matrix.

\[ r_A = \sum d_{AB} \quad \text{(Equ. ii)} \]

A and B in equation ii represents two different taxa. The transformed \( r \)-value (\( r' \)) is required to calculate the distance of an individual taxon from its node.

\[ r'_A = \frac{r_A}{n-2} \quad \text{(Equ. iii)} \]

Where \( n \) is the total number of taxa. Let’s assume A and B form a node called U, the distance of individual taxa (taxa A or B) from the node U can be calculated using the following equation:

\[ d_{AU} = \frac{d_{AB} + (r'_A - r'_B)}{2} \quad \text{(Equ. iv)} \]

In the NJ method, before the construction of the tree, all the given taxa are collapsed into a star tree. Then the pair of taxa having the shortest distance in the matrix is separated from the star and built into a single composite taxon with a node. After the first node is constructed, the newly created taxon is considered as a single taxon, allowing the next most closely related taxon to be joined next to the first node which will create the second node for a new taxon. The cycle is iterated until all internal nodes are resolved. This whole process is also known as star decomposition because it starts with the star formation and decomposition of that star into a tree diagram based on the distances between taxa. NJ method produces an unrooted tree; outgroups are determined on the basis of external knowledge.

Consider the following example of tree construction using the Neighbour-Joining method. For this example, the same pairwise distance matrix given in Figure 9 is used. The first step in the NJ method is the calculation of \( r \) and \( r' \) value using equation ii and iii.

The \( r \)-value is calculated as:

- \( r_A = AB+AC+AD = 2+3+7 = 12 \)
- \( r_B = BA+BC+BD = 2+5+8 = 15 \)
- \( r_C = CA+CB+CD = 3+5+6 = 14 \)
- \( r_D = DA+DB+DC = 7+8+6 = 2 \)

The \( r' \) value is calculated as:

- \( r'_A = \frac{r_A}{n-2} = \frac{12}{4-2} = \frac{12}{2} = 6 \)
\[ r_B' = \frac{r_B}{n-2} = \frac{15}{4-2} = \frac{15}{2} = 7.5 \]
\[ r_C' = \frac{r_C}{n-2} = \frac{14}{4-2} = \frac{14}{2} = 7 \]
\[ r_D' = \frac{r_D}{n-2} = \frac{21}{4-2} = \frac{21}{2} = 10.5 \]

The corrected distances are calculated using equation i to create a new distance matrix (Figure 12) as:

\[ d'_{AB} = 2 – \frac{1}{2}*(12 + 15) = 2 – \frac{27}{2} = 2 – 13.5 = –11.5 \]
\[ d'_{AC} = 3 – \frac{1}{2}*(12 + 14) = 3 – \frac{26}{2} = 3 – 13 = –10 \]
\[ d'_{AD} = 7 – \frac{1}{2}*(12+21) = 7 – \frac{33}{2} = 7 – 16.5 = –9.5 \]
\[ d'_{BC} = 5 – \frac{1}{2}*(15 + 14) = 5 – \frac{29}{2} = 5 – 14.5 = –9.5 \]
\[ d'_{BD} = 8 – \frac{1}{2}*(15 + 21) = 8 – \frac{36}{2} = 8 – 18 = –10 \]
\[ d'_{CD} = 6 – \frac{1}{2}*(14 + 21) = 6 – \frac{35}{2} = 6 – 17.5 = –11.5 \]

![Distance matrix](image.png)

Figure 12: Distance matrix created using the corrected distances for NJ tree construction

The next step in the NJ method is the grouping of taxa, before this, all possible nodes are collapsed to form a star, then the taxa get separated step by step from this star diagram according to their distances (this is why the NJ method is also known as star deformation). According to the distance matrix in Figure 12, A and B taxa are at shortest distance, so these taxa are separated from the star and are grouped together to form a single composite taxon with a node “U” in the middle of both taxon (Figure 13).
The branch length of A and B to the node U is calculated using equation iv as:

\[ d_{AU} = \frac{[2 + (6 - 7.5)]}{2} = \frac{[2 + (-1.5)]}{2} = 0.5/2 = 0.25 \]

\[ d_{BU} = \frac{[2 + (7.5 - 6)]}{2} = \frac{[2 + 1.5]}{2} = 3.5/2 = 1.7 \]

The branch lengths on the NJ phylogenetic tree is adjusted taking into account the new branch length distances (Figure 14).

New composite taxon allows the construction of reduce matrix. Start with actual distances and again construct distance matrix for other taxa (C, D).

\[ d_{CU} = \frac{[(d_{AC} - d_{UA}) + (d_{BC} - d_{UB})]}{2} = d_{CU} = \frac{[(3 - 0.25) + (5 - 1.75)]}{2} = \frac{[2.75 + 3.25]}{2} = 6/2 = 3 \]

\[ d_{DU} = \frac{[(d_{AD} - d_{UA}) + (d_{BD} - d_{UB})]}{2} = d_{DU} = \frac{[(7 - 0.25) + (8 - 1.75)]}{2} = \frac{[6.75 + 6.25]}{2} = 13/2 = 6.5 \]

The new \( r \)-values and the \( r' \)-values are calculated using the new composite distance matrix (Figure 15). The corrected distances are calculated using this distance matrix and a new distance matrix with correct evolutionary distances between taxa is obtained (Figure 15).
According to figure 15, the taxon C have the shortest distance to node U. So, the taxon C is grouped with A-B (it creates new composite taxon A-B-C). A new node “V” is created from U to C. Before grouping of the taxa, the branch lengths are first calculated (Figure 16).

\[ d_{CV} = \frac{[3 + (9 - 9.5)]}{2} = \frac{[3 + (-0.5)]}{2} = \frac{2.5}{2} = 1.25 \]

\[ d_{DV} = \frac{[3 + (9.5 - 9)]}{2} = \frac{[3 + 0.5]}{2} = \frac{3.5}{2} = 1.75 \]

**Optimality-Based Methods**

Clustering-based algorithms only produce a single tree as an outcome. However, it is not possible to determine how this tree is compared with other alternate trees. The optimality-based method has a well-defined algorithm which compares all possible tree topologies and selects a tree that best fits the evolutionary distance matrix. This method is exhaustive and has slow computation. There are two types of algorithms in optimality-based method Fitch–Margoliash (FM), Minimum Evolution (ME) [13,14].
Fitch-Margoliash

The Fitch-Margoliash approach selects the best tree among all available trees based on the minimal deviation between the distances calculated in the overall branches in the tree and the distances in the original dataset. It starts by clustering two taxa with the shortest distance in a single node and temporarily combining all other taxa into a group (Figure 17). It then creates an unrooted 3 taxa tree. Three algebraic equations are solved to determine the branch lengths and distances. The clustering to two taxa helps create a new reduced matrix, and this process is iterated until the tree is entirely resolved. This method selects the best tree among all tree topologies that has the lowest squared deviation of actual distances and calculated tree branch lengths [15]. The criterion of optimality is described by the following formula:

\[
E = \sum_{a=1}^{T-1} \sum_{b=a+1}^{T} \frac{(d_{ab} - p_{ab})^2}{d_{ab}^2}
\]

Where \( E \) is the error of the estimated tree fitting the original data, \( T \) is the number of taxa, \( d_{ab} \) is the pairwise distance between A and B taxa in the original dataset, and \( p_{ab} \) is the corresponding tree branch length [1].

\[ T_a = a + 1 \]

Figure 17: Sample Fitch-Margoliash tree. “a, b and c” represent branch lengths

The distances are calculated using the following algebraic equations:

\[
\begin{align*}
d_{AB} &= a + b \\
d_{AC} &= a + c \\
d_{BC} &= b + c
\end{align*}
\]

The branch lengths are calculated using the following equations:

\[
\begin{align*}
a &= (d_{AB} + d_{AC} - d_{BC})/2 \\
b &= (d_{AB} + d_{BC} - d_{AC})/2 \\
c &= (d_{AC} + d_{BC} - d_{AB})/2
\end{align*}
\]
Consider the following example of tree construction using the Fitch-Margoliash algorithm; The tree construction starts by constructing a distance matrix by calculating dissimilarities between sequences just like done in previous examples. Two taxa with the smallest distance are clustered together in a single node, and all other taxa are temporarily combined into a single group.

Figure 18: Fitch-Margoliash tree construction using a distance matrix. A and B are at the shortest distance so they are clustered into a single node, and all other taxa are grouped temporarily.

Since we combined taxa C, D and E into a single group we need to find the distance of taxa A and B to C-D-E (Figure 19) in order to calculate the branch lengths.

\[ d_{A-CDE} = \frac{(d_{AC}) + (d_{AD}) + (d_{AE})}{3} = \frac{7 + 11 + 12}{3} = \frac{30}{3} = 10 \]

\[ d_{B-CDE} = \frac{(d_{BC}) + (d_{BD}) + (d_{BE})}{3} = \frac{9 + 10 + 10}{3} = \frac{29}{3} = 9.667 \]

Figure 19: Reduced distance matrix created by calculating distance of A and B to C-D-E.
Now using the newly created reduced distance matrix the branch lengths are calculated as follow:

\[ a = \frac{(d_{AB} + d_{A-CDE} - d_{B-CDE})}{2} = \frac{2 + 10 - 9.667}{2} = 1.1665 \]

\[ b = \frac{(d_{AB} + d_{B-CDE} - d_{A-CDE})}{2} = \frac{2 + 9.667 - 10}{2} = 0.8335 \]

\[ cde = \frac{(d_{A-CDE} + d_{B-CDE} - d_{AB})}{2} = \frac{10 + 9.667 - 2}{2} = 8.8335 \]

Figure 20: Phylogenetic tree with actual branch lengths constructed using Fitch-Margoliash algorithm

Now we can treat taxa A and B as a single group and separate C-D-E to create a new distance matrix (Figure 21). The taxa with the shortest distance are selected again from the matrix and their branch length is calculated.

Taxa D-E are at the shortest distance in the new distance matrix so they are clustered in a single node and all other taxa are grouped together (A-B-C). The distance of D and E to the newly created group is calculated as follow:
\[ d_{D-ABC} = \frac{(d_{AD}) + (d_{BD}) + (d_{CD})}{3} = \frac{11 + 10 + 5}{3} = \frac{26}{3} = 8.667 \]

\[ d_{E-ABC} = \frac{(d_{AE}) + (d_{BE}) + (d_{CE})}{3} = \frac{12 + 10 + 5}{3} = \frac{27}{3} = 9 \]

Figure 22: Taxa D and E are clustered into a single node (left). Reduced distance matrix created by calculating the distance of D and E to A-B-C (right)

Now the branch lengths are calculated using the newly created distance matrix

\[ d = \frac{(d_{D-ABC} + d_{DE} - d_{E-ABC})}{2} = \frac{8.667 + 4 - 9}{2} = \frac{3.667}{2} = 1.8335 \]

\[ e = \frac{(d_{E-ABC} + d_{ED} - d_{D-ABC})}{2} = \frac{4 + 9 - 8.667}{2} = \frac{4.333}{2} = 2.1665 \]

\[ abc = \frac{(d_{D-ABC} + d_{E-ABC} - d_{DE})}{2} = \frac{8.667 + 9 - 4}{2} = \frac{13.667}{2} = 6.8335 \]

Figure 23: Phylogenetic tree with actual branch lengths

Only the C taxon is left so to calculate its branch length taxa A-B and D-E are combined to create a new reduced matrix (Figure 24)
The branch length is calculated using the new distances obtained in reduced matrix

\[ ab = (d_{AB-C} + d_{AB-DE} - d_{C-DE})/2 = 8 + 10 - 5 / 2 = 13.75 / 2 = 6.875 \]
\[ c = (d_{AB-C} + d_{DE-C} - d_{AB-DE})/2 = 8 + 5 - 10.75 / 2 = 2.25 / 2 = 1.125 \]
\[ de = (d_{AB-DE} + d_{DE-C} - d_{AB-C})/2 = 10.75 + 5 - 8 / 2 = 7.75 / 2 = 3.875 \]
Now to calculate the branch length of a in figure 26 subtract the average of A and B \((A+B/2 = 1.1665+0.8335/2 = 2/2 = 1)\) from the branch length of AB \((a = [(AB) – (A+B/2)] = 6.875 – 1 = 5.875)\). Similarly, for b the average of D and E \((D+E/2 = 1.8335+2.1665/2 = 4/2 = 2)\) is subtracted from branch length DE \((b = [(DE) – (D+E/2)] = 3.875 – 2 = 1.875)\). Adding these final branch lengths, a complete phylogenetic tree with correct evolutionary distances is obtained (Figure 27).

Minimum Evolution

Minimum evolution constructs a tree with a similar method but uses a different criterion of optimality that finds a tree with a minimum total branch length among all possible trees. The following formula describes the criterion of optimality:

\[
S = \sum b_a
\]

Where S is the sum of branch lengths for given tree topologies and \(b_a\) is the \(a\)th branch length. The theoretical basis of the minimum evolution method is the statistical evidence presented by A. Rzhetsky and M. Nei in 1993 [16] that shows that when unbiased estimates of
evolutionary distances are used, regardless of the number of sequences, the predicted value of $S$ becomes smaller for true topology.

**Character-Based Methods**

The character-based methods compare all sequences simultaneously considering one character/site at a time. Character-based methods include maximum parsimony and maximum likelihood. These approaches use probability and take into account the sequence variations [17]. Both methods take the tree with the highest score into consideration, which needs a minimum number of changes to align.

**Maximum Parsimony**

Maximum Parsimony (MP) is the most common and widely accepted tree construction approach to date [18]. Because it uses a character-based algorithm, this approach is different from the distance-based approaches previously discussed. This approach works by scanning through all potential tree structures and assigning a cost to each tree. Parsimony is based on the presumption that the most possible tree is the one that needs the least number of changes to explain the alignment data [19]. The premise that taxa or nodes are sharing a common characteristic does so because it inherited that attributes from a common ancestor [20].

Conflicts with this approach are explained under the term homoplasy. There are three ways of reserving conflicts: reversal (return to the original state), convergence (unrelated taxa evolved the same characteristic completely independently) and parallelism (different taxa may have similar mechanisms that develop a character in a certain way) [19,20]. The tree with a lowest tree score or length, as determined by the number of changes accumulated along the branches, is called the most parsimonious tree, and it best represents the evolutionary pattern [20,21].

Maximum Parsimony is also different from other approaches as it does not find the branch lengths but the total overall length in terms of the number of changes. Maximum Parsimony often considers two or more trees equal and provides no direct answer on which tree is the actual evolutionary tree. A strict (majority rule) consensus is used to overcome this problem in most situations.

Traditional Parsimony method uses recursion to find the minimum amount of change within the trees. This is done by starting at the leaf of a tree and working up towards the root. It is known as post-order transversal [18,20]. Whereas the other approach, Weighted Parsimony, gives the algorithm a cost factor and weights certain scenarios accordingly.

Often an artefact known as the Long-Branch Attraction exists in parsimony approach, and it should be handled. The length of the branch shows the number of substitutions between taxa or nodes. Parsimony assumes that all taxa evolve at the same rate and contribute that same
amount of information [20]. Long-branch is the phenomena in which rapidly evolving taxa are grouped together on a tree when they have numerous mutations (Figure 28). Anytime two long branches are present, they may be attracted to one another [20].

Figure 28: Long-Branch attraction in Maximum Parsimony method. Branches leading to A and C might be expected to have a higher number of character state transformations than the branches leading to B and D.

Maximum Likelihood

Maximum Likelihood method was proposed by Felsenstein in 1981 [22]. This is one of the most computationally intensive methods, but it is most flexible. ML uses probabilistic models to choose a best tree that has the highest probability or likelihood of reproducing the observed data [1]. ML trees represent the most accurate evolutionary processes. This approach considers every single position in the alignment data to search for all possible tree topologies and not only the informative sites. ML calculates the total probability of ancestral sequences evolving to internal nodes and ultimately to current sequences by using a specific substitution model that has
probability values of residue substitutions. For instance, for DNA sequences using the Jukes-Cantor model, the probability (P) that a nucleotide remains the same after the time (t) is:

\[ P(t) = \frac{1}{4} + \frac{3}{4}e^{-\alpha t} \]

Where \( \alpha \) is the nucleotide substitution rate which is either empirically allocated or calculated from the raw datasets.

The ability to render statistical associations between topologies and data sets is one of ML's major advantages. Maximum Likelihood makes predictions that the model used is accurate and that the approach is inconsistent if the model does not correctly represent the underlying data set. A downside of ML is the rigorous computation, and recent research indicates that a given phylogenetic tree may have multiple maximal probability points [23].

Table 1: Comparison of phylogenetic tree construction methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPGMA</td>
<td>Fast and reliable for related sequences</td>
<td>Evolution rate is constant in all branches</td>
</tr>
<tr>
<td>Neighbour-Joining</td>
<td>Faster than the character-based method</td>
<td>Conversion from sequence data to distance data leads to loss of information</td>
</tr>
<tr>
<td>Fitch-Margoliash</td>
<td>Less sensitive to variations in evolutionary rate</td>
<td>Dependent on the model used to obtain the distance matrix</td>
</tr>
<tr>
<td>Maximum Parsimony</td>
<td>Fast, robust and appropriate for very similar sequences and a small number of sequences</td>
<td>Performance is not satisfactory and very time consuming</td>
</tr>
<tr>
<td>Maximum Likelihood</td>
<td>Suitable for very dissimilar sequences and phylogeny is clear</td>
<td>Slow and computationally intensive</td>
</tr>
</tbody>
</table>

Conclusion

Major sequence alignment and phylogenetic tree construction methods are discussed in this paper with their pros and cons. Global and Local sequence alignment algorithms based on dynamic programming were analyzed to compare two biological sequences to find out similarities between them. The goal of pairwise sequence alignment is to establish algorithms and bioinformatics tools to produce optimal alignment between the sequences. Phylogenetic trees were constructed using multiple sequence alignment data. These algorithms are programmed into different languages to construct phylogenetic trees. UPGMA and Neighbour-Joining are computationally fast; however, these methods provide predictions to find the best trees. On the other hand, Fitch-Margoliash & Minimum Evolution has more accuracy, but they can be computationally exhaustive. Maximum Parsimony and Maximum Likelihood are more accurate than the distance-based methods but not suggested with big sequence data.
References