Minimising branch crossings in phylogenetic trees

Sung-Hyuk Cha*

Department of Computer Science,
Pace University,
New York, NY, USA
Email: scha@pace.edu
*Corresponding author

Yoo Jung An

Engineering Technologies and Computer Sciences,
Essex County College,
Newark, NJ, USA
Email: yan@essex.edu

Abstract: While phylogenetic trees are widely used in bioinformatics, one of the major problems is that different dendrograms may be constructed depending on several factors. Albeit numerous quantitative measures to compare two different phylogenetic trees have been proposed, visual comparison is often necessary. Displaying a pair of alternative phylogenetic trees together by finding a proper order of taxa in the leaf level was considered earlier to give better visual insights of how two dendrograms are similar. This approach raised a problem of branch crossing. Here, a couple of efficient methods to count the number of branch crossings in the trees for a given taxa order are presented. Using the number of branch crossings as a fitness function, genetic algorithms are used to find a taxa order such that two alternative phylogenetic trees can be shown with semi-minimal number of branch crossing. A couple of methods to encode/decode a taxa order to/from a chromosome where genetic operators can be applied are also given.

Keywords: clustering; dendrogram; hierarchical clustering; phylogenetic tree; visualisation.

Reference to this paper should be made as follows: Cha, S-H. and An, Y.J. (2016) 'Minimising branch crossings in phylogenetic trees', *Int. J. Applied Pattern Recognition*, Vol. 3, No. 1, pp.22–37.

Biographical notes: Sung-Hyuk Cha received his PhD in Computer Science from State University of New York at Buffalo in 2001 and BS and MS degrees in Computer Science from Rutgers, the State University of New Jersey in 1994 and 1996, respectively. From 1996 to 1998, he was working in the area of medical information systems such as PACS, teleradiology, and telemedicine at Information Technology R&D Center, Samsung SDS. During his PhD years, he was affiliated with the Center of Excellence for Document Analysis and Recognition (CEDAR). His main interests include computer vision, data mining, pattern matching and recognition.

Yoo Jung An is an Assistant Professor of Computer Sciences at Essex County College (ECC), Newark, New Jersey. She is currently serving as a program coordinator of the Computer Science and Computer Information Systems programs that are offered by the Division of Engineering Technologies and Computer Sciences at ECC. She received her MS and PhD degrees in Computer Science from New Jersey Institute of Technology (NJIT) in 2004 and 2008, respectively. Her research interests include Semantic Web, ontologies, health informatics, deep web, data warehouses, artificial intelligence and internet security. Her research has been published in *International Journal of Engineering and Technology*, *International Journal of Computational Models and Algorithms in Medicine*, *IEEE Computer*, *Communications of the International Information Management Association*, and many conference proceedings.

1 Introduction

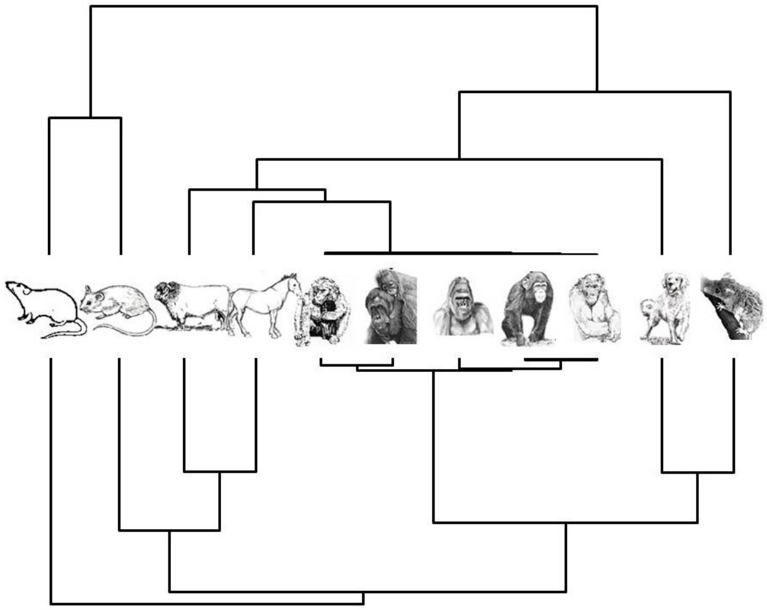
Hierarchical clustering methods have been often adopted to construct phylogenetic trees in biological taxonomy and bioinformatics (Dunn and Everitt, 1982). They produce a visual tree representation of the hierarchical clusters called a *dendrogram* which may correspond to a *phylogenetic tree* revealing possible patterns of evolution in bioinformatics and taxonomy. Numerous methods to construct a dendrogram have been studied (see Dunn and Everitt, 1982; Duda et al., 2000; for details of methods).

While phylogenetic trees are widely used in bioinformatics, different or even conflicting dendrograms are produced depending on several factors. First, taxa representation such as amino acid sequences in proteins, nucleotide sequences in nucleic acids, or certain specific gene sequences affects dramatically on the shape of dendrogram. For example, genes for myoglobin and alpha haemoglobin may have evolved independently within each evolutionary line of animals (Dunn and Everitt, 1982). Other factors impactious to the phylogenetic tree structure include which pairwise proximity measure between taxa is used and how the distance between clusters is defined.

Comparing multiple conflicting dendrograms is of great interest and several quantitative measures including the earliest *cophenetic correlation coefficient* (Sokal and Rohlf, 1962) appear in literature (Robinson and Foulds, 1981; Nye et al., 2005). However, it is a very hard and subjective problem and thus needs for visual comparison of multiple phylogenetic trees were raised (Nye et al., 2005; Amenta and Klingner, 2002).

Finding a proper order of leaf nodes (taxa) plays a salient role in visualising trees (Bar-Joseph et al., 2001; Dwyer and Schreiber, 2004). Two conflicting dendrograms are often visualised side by side in almost aligned leaf node orders where the number of crossings between leaf node orders is minimised (Zainon and Calder, 2006; Scornavacca et al., 2011) which is an NP-hard problem (Scornavacca et al., 2012). In Cha (2013), visualising two dendrograms in a fixed leaf node order as shown in Figure 1 was proposed. *Branch crossing* is inevitable in many applications. Here, two methods to count the number of branch crossing are presented.

Figure 1 Sample two phylogenetic trees on a fixed taxa order



The problem of finding an appropriate order of taxa where the number of branch crossings in both dendrograms is minimised is considered. *Genetic algorithms*, which provides good solutions to many optimisation problems (Goldberg, 1989; Mitchell, 1996) are useful to the branch crossing minimisation problem. A couple of methods to encode/decode taxa order to/from a *chromosome* are presented so that genetic operators such as *mutation* and *crossover* can be applied.

The rest of the paper is organised as follows. The preliminary Section 2 defines basic terminologies and notations and describes how the dendrogram is represented and visualised. In Section 3, algorithms to count the number of branch crossings and drawing dendrograms with branch crossing are presented. Section 4 presents genetic algorithms using the number of branch crossing as a performance measure to find the order of taxa with semi minimum number of branch crossing. Three experimental case studies are given in Section 5. Finally, Section 6 concludes this work with future works.

2 Phylogenetic tree representation

Before embarking on the branch crossing minimisation algorithms, it is necessary to understand how the phylogenetic tree is represented and visualised. This preliminary section provides a brief description of the constructing and visualising phylogenetic tree algorithms and defines terminologies and notations used in this article.

Consider a set of five taxa $S = \{A, B, C, D, E\}$ and let n be the number of taxa, $n = |S| = 5$. These taxa could be different species of interest, which are also referred to as *operational taxonomic units* or simply OTUs in numerical taxonomy. These taxa can be represented by nucleotide sequence, protein sequence, or characteristic feature vector, etc. When an appropriate pairwise distance measure is used depending on taxa representation, a distance matrix, D among taxa is computed. Different or sometimes controversial phylogenetic trees are produced based on choices of taxa representation and/or pairwise distance measure.

Consider a sample distance matrix of all five taxa is given in Table 1. A distance matrix, D is typically an input to the bottom-up hierarchical clustering algorithms which are widely used in building phylogenetic trees.

Table 1 A sample distance matrix of five taxa, D

	A	*B*	*C*	*D*	*E*
A	0	8.0	20.0	16.0	9.8
B	8.0	0	21.5	17.9	9.8
C	20.0	21.5	0	4.0	11.7
D	16.0	17.9	4.0	0	8.1
E	9.8	9.8	11.7	8.1	0

Although there are several bottom-up hierarchical clustering algorithms (see Dunn and Everitt, 1982; Duda et al., 2000) for a variety of methods and their descriptions), a sample output tree representation, T, useful for the later algorithms to count the number of branch crossings, is given in Table 2 which uses the *agglomerative single linkage clustering* method.

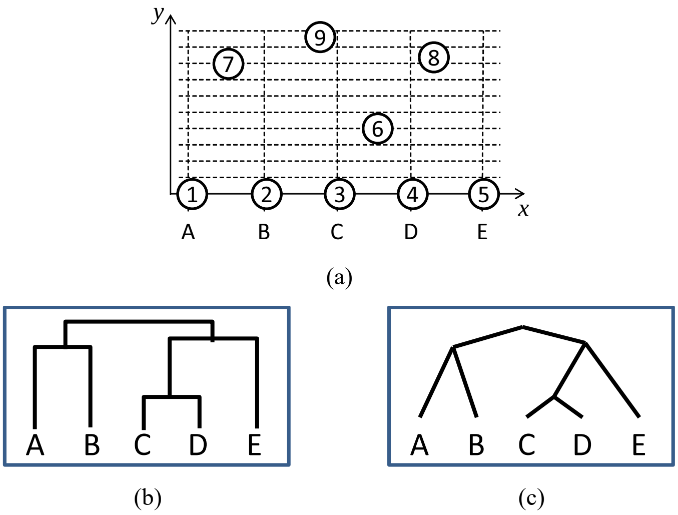
Table 2 A sample dendrogram representation, T

Cluster	*N*	C_l	C_r	*x*	*y*
$\{A\}$	1	-	-	1	0
$\{B\}$	2	-	-	2	0
$\{C\}$	3	-	-	3	0
$\{D\}$	4	-	-	4	0
$\{E\}$	5	-	-	5	0
$\{C, D\}$	6	3	4	3.5	4.0
{A, B}	7	1	2	1.5	8.0
$\{C, D, E\}$	8	6	5	4.25	8.1
$\{A, B, C, D, E\}$	9	7	8	2.88	9.8

The height of Table 2 or dendrogram, T in Table 2 is $2n - 1$. The first $n = 5$ rows correspond to the leaf level nodes where each row is a taxon. All leaf nodes must be in the same bottom level. The remaining $n - 1 = 4$ rows are internal nodes representing clusters of two or more taxa. An internal node may be considered as a possible common ancestor of all its descendants. Each node, N_x is often treated as a cluster and is assigned to a countable integer value. Each internal node has exactly two child denoted as C_l and C_r. Let $sib(N_x)$ denote a sibling node of N_x, e.g., sib(4) = 3. Let $N_x.C_l$ and $N_x.C_r$ denote two child nodes of N_x, e.g., $8.C_l = 6$. All nodes except for the root node have exactly one parent node. Let $par(N_x)$ denote the parent node of N_x, e.g., $par(6) = 8$.

Each node has x and y cartesian coordinate position values where x and y are horizontal and vertical axis values, respectively as shown in Figure 2(a). Either *phenogram* or *cladogram* can be drawn by connecting parent and their children nodes as shown in Figure 3(b) and Figure 3(c), respectively.

Figure 2 Geometry of a dendrogram, (a) cluster node in Cartesian coordinate system (b) phenogram (c) cladogram (see online version for colours)



The *phenogram* of T denoted as *phe*(T) is a set of line segments where parent and child nodes are connected with a right angle segments and often called *u-shaped line dendrogram*. *phe*(T) is defined in equation (4) where N_I is an internal node in T and has three line segments: one beam and two column line segments as defined in equations (1), (2) and (3), respectively.

$$beam(N_I) = \big((N_I.C_l.x,\ N_I.y), (N_I.C_r.x,\ N_I.y)\big) \tag{1}$$

$$col(N_I.C_l) = \big((N_I.C_l.x,\ N_I.C_l.y), (N_I.C_l.x,\ N_I.y)\big) \tag{2}$$

$$col(N_I.C_r) = \big((N_I.C_r.x,\ N_I.C_r.y), (N_I.C_r.x,\ N_I.y)\big) \tag{3}$$

$$phe(T) = \bigcup_{N_I \in T} \{beam(N_I), col(N_I, C_l), col(N_I, C_r)\} \tag{4}$$

The *cladogram* of T denoted as *cld*(T) is a set of line segments where parent and child nodes are connected with a straight line segment called leg. *cld*(T), often called *v-shaped line dendrogram*, is defined in equation (7) where N_I is an internal node in T and has two line segments (legs) as defined in equations (5) and (6).

$$leg(N_I.C_l) = \big((N_I.C_l.x,\ N_I.C_l.y), (N_I.x,\ N_I.y)\big) \tag{5}$$

$$leg(N_I.C_r) = \big((N_I.C_r.x,\ N_I.C_r.y), (N_I.x,\ N_I.y)\big) \tag{6}$$

$$cld(T) = \bigcup_{N_I \in T} \{leg(N_I.C_l), leg(N_I, C_r)\} \tag{7}$$

The bottom-up hierarchical clustering algorithms in this paper take D and O as inputs where O is an order of taxa, e.g., O = (1, 2, 3, 4, 5) in Table 2. A pseudo-code of an algorithm to construct T is described in Algorithm 1.

Algorithm 1 Conventional centre method

```
for each leaf node, $N_x = 1 \sim n$,                              1
    $N_x.x = idx(N_x)$                                             2
    $N_x.y = 0$                                                    3
for each internal node, $N_x = n + 1 \sim 2n - 1$,                 4
    $c_1$ and $c_2$ = the closest pair in $D$                      5
    $Nx.cl = c1$ and $Nx.cr = c2$                                  6
    $Nx.y = D(c_1, c_2)$                                           7
    $Nx.x = (c_1.x + c_2.x) = 2$                                   8
    delete $c_1$ and $c_2$ entries from $D$                        9
    add $N_x$ to $D$ using a distance bw clusters.                 10
```

First in lines 1 ~ 3, the index of taxa, N_x in O corresponds to $N_x.x$. Let $idx(N_x)$ be the position of N_x in O. The remaining internal nodes are created by merging two other nodes. The closest pair of two nodes in D, c_1 and c_2 are merged to a new parent node, N_x and its height, $N_x.y$ is the distance between two children nodes in D in line 7. $N_x.x$ is obtained simply by taking the centre of two children nodes in line 8. Finally, rows and columns of c_1 and c_2 are merged into a single row and column for N_x entries in D.

In the process of merging two nodes into one, distances between the new node and the remaining nodes need to be calculated. Several definitions of distance between clusters are in use, i.e., *single*, *complete*, *group average*, *centroid*, *ward*, etc. Quite significantly different dendrograms are produced depending on the choice of distance between two clusters.

3 Phylogenetic trees with branch crossing

This section first considers a problem of visualising Phylogenetic Trees with branch crossing avoiding the *segment overlapping* problem and then provides algorithms to count the number of branch crossings in a given taxa order. There are $n!$ possible orders of permutation of n taxa and only 2^{n-1} number of taxa orders do not have branch crossing but the branch crossing is inevitable in the rest of taxa orders (Cha, 2013).

For the example of five taxa, $\{A, B, C, D, E\}$ branch crossings occur in orders (3, 5, 1, 4, 2), (2, 5, 1, 3, 4) and (2, 4, 1, 5, 3) as shown in Figure 3. When algorithm ?? is used, dendrograms in Figure 3(b) and Figure 3(c) are displayed poorly due to segment overlaps as well as branch crossings. This may be a reason that visualising dual trees by minimising crossings between two different taxa orders have been used widely such as in Amenta and Klingner (2002), Bar-Joseph et al. (2001), Dwyer and Schreiber (2004), Zainon and Calder (2006) and Scornavacca et al. (2011) rather than visualising them in a single fixed taxa order.

The problem overlapping segments in a dendrogram can be resolved in two different paradigms. An internal node, $N_I.x$ can be placed on the *beam*(N_I) which is between $N_I.C_l.x$ and $N_I.C_r.x$ instead of the centre as in the line 8 in Algorithm 1. The first paradigm is to place $N_I.x$ such that the number of branch crossing is minimised as shown in Figure 4(a) and Figure 4(b) which correspond to ones in Figure 3(b) and Figure 3(c), respectively. This concept called min-crossing method was introduced in Cha (2013) as an ongoing research. The second paradigm is to place $N_I.x$ as close to the centre of *beam*(N_I) as possible while avoiding the segment overlaps slightly by as shown in Figure 4(c) and Figure 4(d). This second approach called *α-centric* may introduce more branch crossings, i.e., 2 vs. 3 in taxa order (2, 5, 1, 3, 4) and 4 vs. 5 in taxa order (2, 4, 1, 5, 3) using two methods as depicted in Figure 4.

Figure 3 Poor dendrograms due to branch crossing and segment overlaps, (a) (3, 5, 1, 4, 2) (b) (2, 5, 1, 3, 4) (c) (2, 4, 1, 5, 3) (see online version for colours)

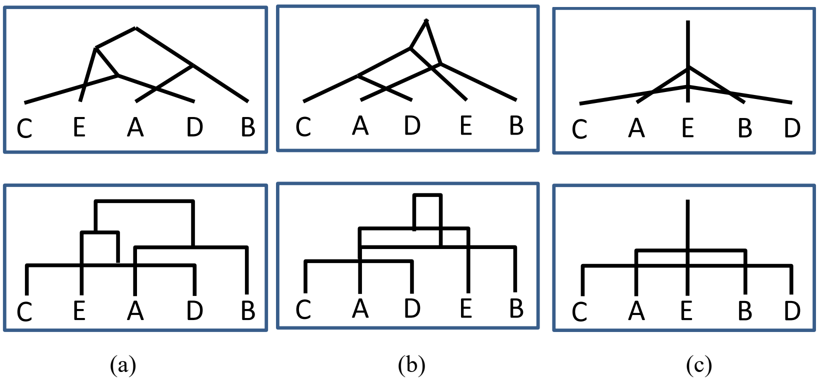
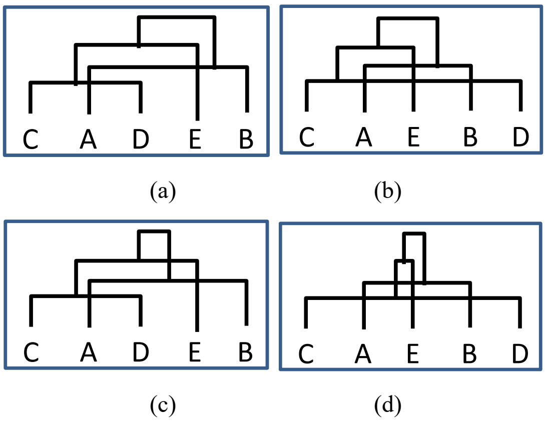
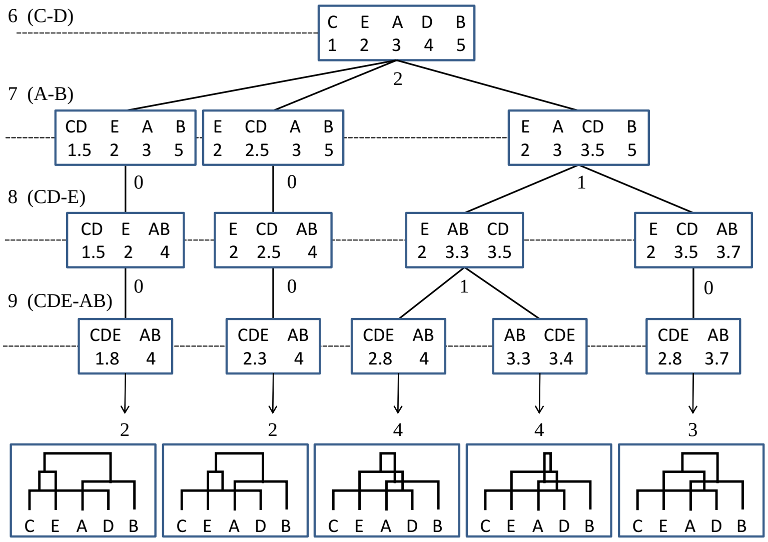


Figure 4 Min-crossing and α-centric methods to resolve segment overlaps, (a) (2, 5, 1, 3, 4) using (b) (2, 4, 1, 5, 3) using the min-crossing method the min-crossing method (c) (2, 5, 1, 3, 4) using (d) (2, 4, 1, 5, 3) using the α-centric method the α-centric method (see online version for colours)



Although the *min-crossing* approach draws a dendrogram with the minimum number of branch crossing without any segment overlapping, the computational complexity of the naïve algorithm illustrated in Figure 5 is high. A naïve algorithm to compute the minimum number of branch crossings starts with merging the closest pair of clusters and counting how many other clusters are between these two clusters. If there are m clusters between them, the merged new cluster can be placed in any one of $m - 1$ places. For an example in Figure 5, the closest clusters are $\{C\}$ and $\{D\}$ and there are two other clusters $\{E\}$ and $\{D\}$ between them. There are two branch crossings between $\{C\}$ and $\{D\}$. When this process is repeated recursively, the leaf node in the traversal tree contains only two clusters with the sums of all crossings. Examining all possible new orders, all possible topologies of the dendrogram can be generated. The number of branch crossing for each topology is the cumulated sum of number of clusters between the closest clusters in the respective path from the root. The dendrogram topology with the minimum number of branch crossings can be selected.

Figure 5 Sketchy illustration of min-crossing method on (3, 5, 1, 4, 2) (see online version for colours)

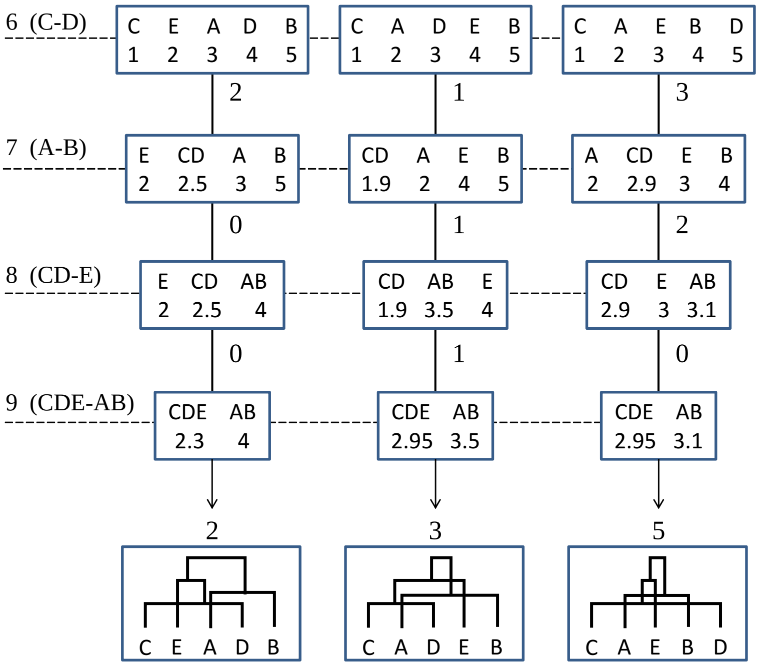


The α-centric method can be computed very efficiently. The order of clusters is based on x coordinate value. Instead of examining all possible position in the new order, the merged cluster N_l is inserted at $(N_l.C_l.x + N_l.C_r.x) / 2$ position. If another node N_z has the same value, value is added or subtracted depending on the orientation of $sib(N_z)$. The α-centric method is illustrated in Figure 6 with three examples. The resulting T for the taxa order (C, A, D, E, B) is given in Table 3.

Table 3 An illustration of α-centric method

Cluster	*N*	C_l	C_r	*x*	*y*
$\{A\}$	1	-	-	2	0
$\{B\}$	2	-	-	5	0
$\{C\}$	3	-	-	1	0
{D}	4	-	-	3	0
{E}	5	-	-	4	0
$\{C, D\}$	6	3	4	1.9	4.0
$\{A, B\}$	7	1	2	3.5	8.0
$\{C, D, E\}$	8	6	5	2.95	8.1
$\{A, B, C, D, E\}$	9	7	8	3.23	9.8

Figure 6 Displaying two dendrograms: T_2 on top and T_3 below (see online version for colours)



4 Genetic algorithm for crossing minimisation

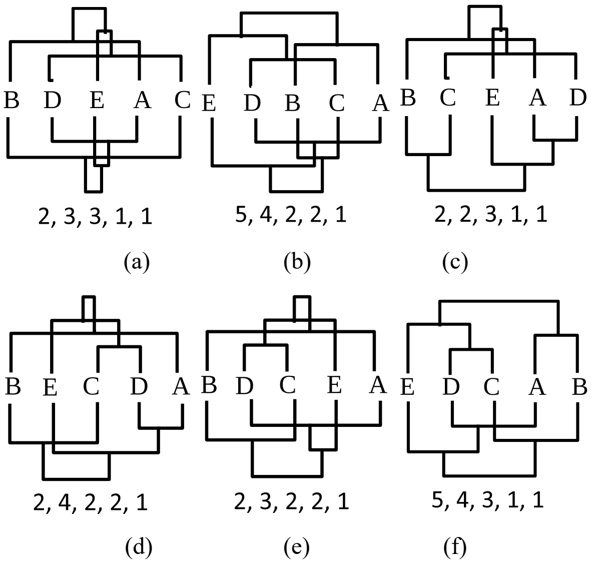
Genetic algorithms which are based on natural processes of evolution and the survival-of-the-fittest concept often provide good solutions to many optimisation problems (Goldberg, 1989; Mitchell, 1996). In order to utilise genetic algorithms to find taxa order such that the number of branch crossing is minimised, taxa order must be encoded so that genetic operators, such as *mutation* and *crossover*, can be applied. In this section, a couple of methods to encode the taxa order to the *artificial chromosome* are presented and compared.

Table 4 encoding/decoding process with an example of $(B, D, E, A, C) \Leftrightarrow (2, 3, 3, 1, 1)$

Taxa order O	*Base order B*	*Chromosome C*
B	$(A, \underline{B}, C, D, E)$	2
D	$(A, C, \underline{D}, E)$	3
E	$(A, C, \underline{E},)$	3
A	$(\underline{A}, C)$	1
C	$(\underline{C})$	1

The first chromosome is a vector of length n where each element in position p can have integer values between 1 and $n - p + 1$. Table 4 illustrates how a taxa order $O = (B, D, E, A, C)$ can be encoded to or decoded from the chromosome $C = (2, 3, 3, 1, 1)$ on the example of five taxa. First the base order, B contains all taxa are sorted in alphabetic order. Starting from the first taxon, t_1 in O, the position of t_1 in P is assigned to represent the taxa. Eliminating t_1 from both O and B, the process is repeated until only one element remains.

Figure 7 Mutation and crossover operation on chromosomes, (a) a sample order (b) a sample order (c) a new order mutated from Figure 7(a) (d) a new order mutated from Figure 7(b) (e) crossover with Figure 7(a) and Figure 7(b) (f) crossover with Figure 7(a) and Figure 7(b)



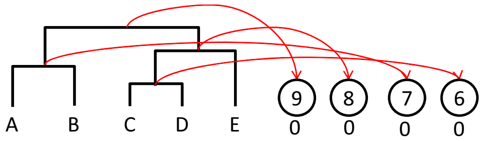
Two popular genetic operations are depicted in Figure 7. The mutation is to change the value of a randomly chosen position on C to another integer value within its allowed range. For example, when the second position value 3 in $C = (2, 3, 3, 1, 1)$ in Figure 7(a) is changed 2, the new taxa order $O = (B, C, E, A, D)$ is obtained shown in Figure 7(c). When the first position value 5 in $C = (5, 4, 2, 2, 1)$ in Figure 7(b) is changed 2, the new taxa order $O = (B, E, C, D, A)$ is obtained shown in Figure 7(d). The crossover operation takes two parent chromosomes and produces two children chromosomes. Consider two parent chromosomes $C_1 = (2, 3, 3, 1, 1)$ and $C_2 = (5, 4, 2, 2, 1)$ in Figure 7(a) and Figure 7(b). When a randomly selected position $p = 2$, the first child chromosome takes the part of C_1 from 1 to p and the remaining part $p + 1$ to n from C_2 as given in Figure 7(e). The second child takes the other parts as shown in Figure 7(f).

The lower number of branch crossing, the higher chance it will survive in genetic algorithms. After long generations later, the population of new generation of chromosomes likely has taxa orders with low number of branch crossings in both dendrograms.

The Search space of the aforementioned chromosome representation is n! and when n is large, it might take very long time before a reasonable taxa order is found. An alternative chromosome representation has its search space 2^{n-1}. While not allowing branch crossing in one dendrogram, it searches the taxa order whose branch crossing on the other dendrogram is minimised. A dendrogram of n taxa can be drawn in 2^{n-1} ways without crossing (Cha, 2013).

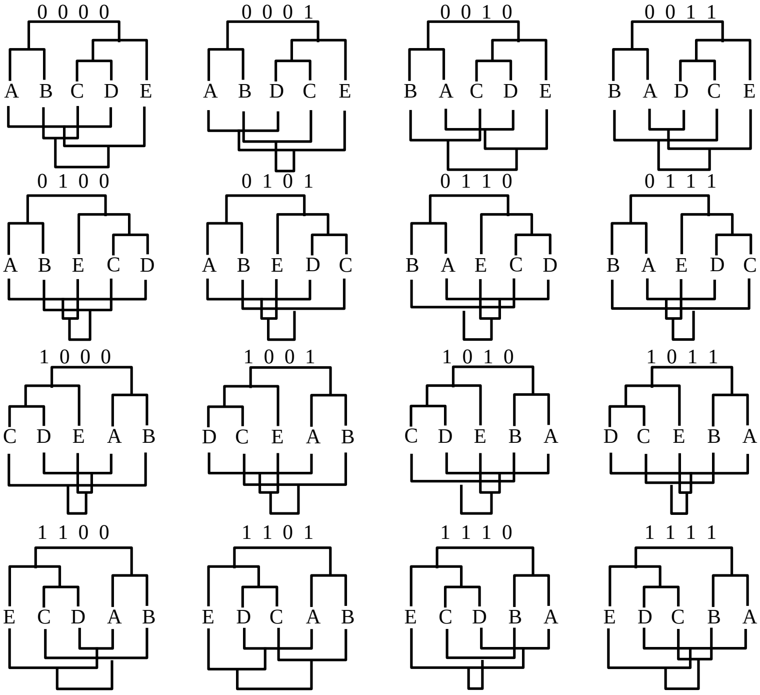
The alternative chromosome representation has a binary vector *I* of size *n* – 1. Each position in *I* corresponds to internal nodes from highest to lowest as depicted in Figure 8.

Figure 8 Internal node chromosome (see online version for colours)



When the binary value of a certain entry in *I* is 1, the left and right children nodes are switched. On the example of five taxa, Figure 9 shows all taxa orders generated by the internal node chromosome. Standard mutation and crossover operators on binary vectors can be applied.

Figure 9 Complete search space of internal node chromosome when *n* = 5



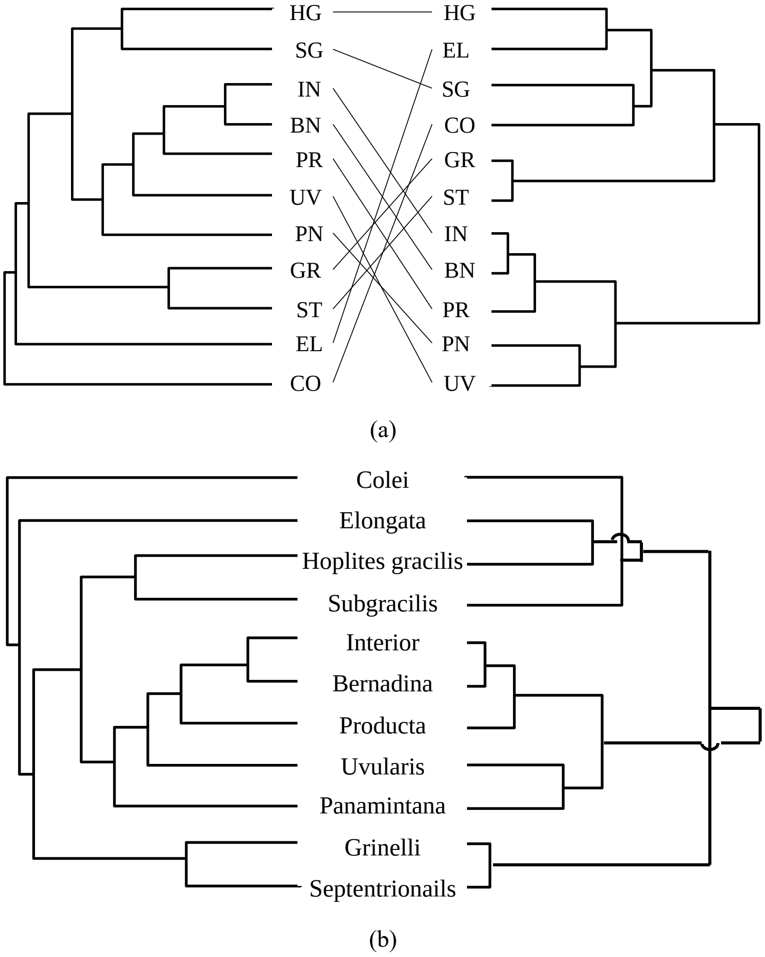
5 Case studies

This section considers three case studies where researchers are interested in comparing multiple phylogenetic trees on the same set of taxa. Using the method presented in previous two sections, two dendrograms are shown together on the the samel taxa order with presumably minimal number of branch crossings.

5.1 Hoplites producta

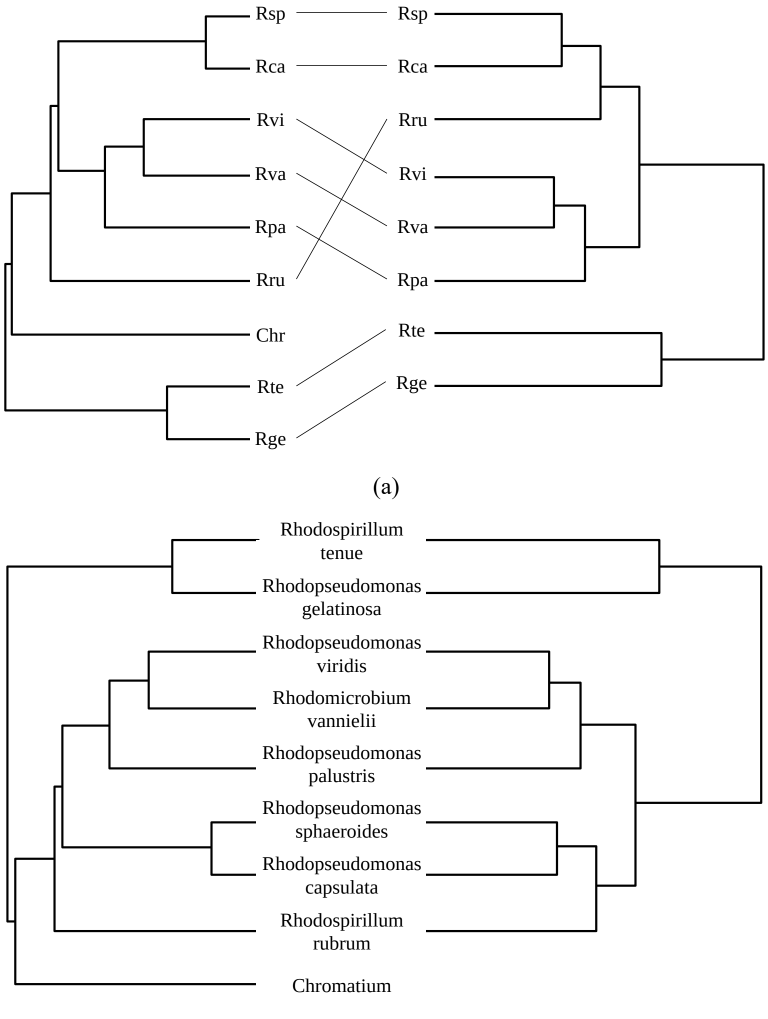
In (15), the relationships among 11 various forms of the bee *Hoplites producta* based on 23 characteristic comparisons were studied. The names and respective abbreviations of *Hoplites* are Hoplites gracilis (HG), Subgracilis (SG), Interior (IN), Bernadina (BN), Panamintana (PN), Producta (PR), Colei (CO), Elongata (EL), Uvularis (UV), Grinelli (GR), and Septentrionails (ST).

Figure 10 Euclidean vs. correlation coefficient proximities, (a) phylogenetic trees T_{eud} and T_{corr} in (15) (b) phylogenetic trees in rearranged taxa order



Two different dendrograms, T_{eud} and T_{corr} using *Euclidean* and *correlation coefficient* proximity measures among taxa were produced as shown in Figure 10 on the left and right sides, respectively. Original dendrograms that appear in Dunn and Everitt (1982) and Michener (1970) are shown side by side in Figure 10(a) with two different taxa orders. While setting no branch crossing on the left dendrogram, the optimal taxa order that minimises the number of branch crossings on the right side dendrogram is found and both dendrograms are constructed on a fixed taxa order as shown in Figure 10(b). Despite a couple of branch crossings, the proposed method visualising a pair of dendrograms provides better visual insights to compare dendrograms.

Figure 11 16 S ribosomal RNA vs. cytochrome c sequence, (a) phylogenetic trees in (16) (b) phylogenetic trees in rearranged leaf nodes



(a)

(b)

5.2 Purple photosynthetic bacter

In Woese et al. (1980), the relationships among various purple photosynthetic bacteria as determined by 16 *S ribosomal RNA* sequence comparisons and *cytochrome c* sequence comparisons were studied on following taxa: Rhodopseudomonas sphaeroides (Rsp), Rhodopseudomonas capsulata (Rca), Rhodopseudomonas viridis (Rvi), Rhodomicrobium vannielii (Rva), Rhodopseudomonas palustris (Rpa), Rhodospirillum rubrum (Rru), Rhodospirillum tenue (Rte), and Rhodopseudomonas gelatinosa (Rge). While a 16 S ribosomal RNA sequence of Chromatium (Chr) is is available, no cytochrome c sequence of it is available in Dunn and Everitt (1982) and Woese et al. (1980).

While two original respective dendrograms were constructed in different taxa orders (Woese et al., 1980) as shown in Figure 11(a), it was concluded that the two dendrograms are remarkably similar, suggesting that gene transfer should not be held responsible for the conflict between the classifications based on sequence data and those obtained by traditional means (Dunn and Everitt, 1982). Yet, it is hard to see the similarities when the taxa orders are different.

Using only the subset of taxa that appear in both dendrograms, it did not take long to find a taxa order with no branch crossing. Only problem here is the remaining taxa which only appear on one of the dendrograms. Since there was only one taxon, Chromatium that appears on only one of the dendrograms, a taxa order such that this non-intersecting taxon can be placed in the beginning or end of the taxa order was luckily found as shown in Figure 11(b). In general, however, treating non-intersecting taxa is one of the future works.

5.3 Forkhead box protein P2

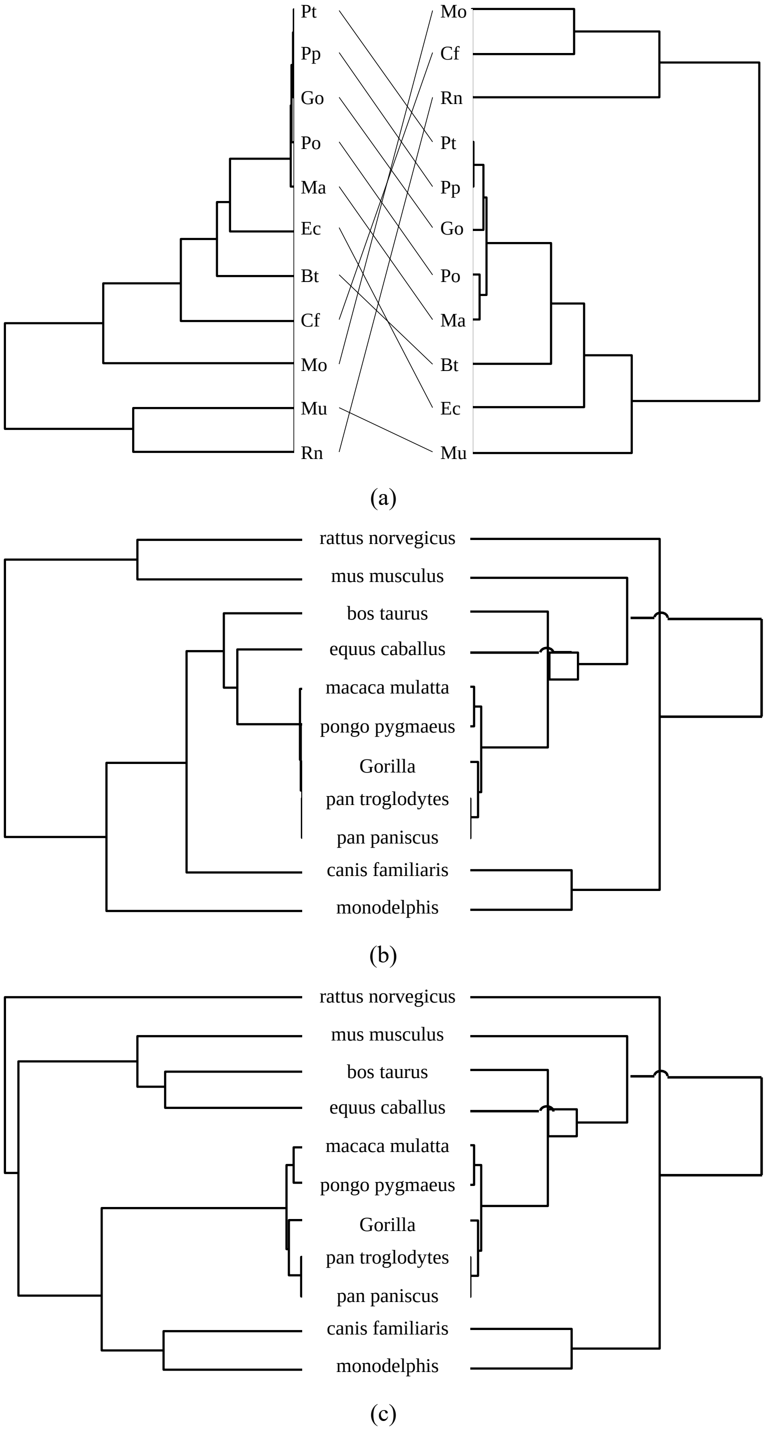
In Cha (2013), *Forkhead box protein P2* or simply *FoxP2* gene DNA sequences which appear in the following 11 species were considered. These species include bos taurus (Bt), canis familiaris (Cf), equus caballus (Ec), gorilla (Go), macaca mulatta (Ma), monodelphis (Mo), mus musculus (Mu), pan paniscus bonobo (Pp), pan troglodytes chimp (Pt), pongo pygmaeus bornean orangutan (Po), and rattus norvegicus (Rn).

Although numerous distinct alternative phylogenetic trees can be produced, three distinct dendrograms T_1, T_2 and T_3 were examined in Cha (2013). The *Jukes-Cantor* method to calculate pairwise distances is used for T_2 and T_3 whereas the *alignment-score* is used for T_1. The score to treat *indels* in nucleotides is used for T_2 and T_3 whereas the *pairwise-delete* is used for T_1. The *single linkage*, *unweighted pair group method with arithmetic mean*, and *complete linkage* clustering methods are used for T_1, T_2, and T_3, respectively. T_1, T_2, and T_3 are all distinct alternative dendrograms.

The original dendrograms for T_1 and T_3 are shown in Figure 12(a) with two different taxa orders. While having no branch crossing on T_1, the genetic algorithm found a taxa order such that the number of branch crossing is only 2 on T_3 as shown in Figure 12(b).

T_1 and T_2 were already given previously in Figure 1 on the top and bottom, respectively with no branch crossing on both sides. The same taxa order founded in T_1 and T_3 is also used in T_1 and T_2. In Figure 12(c), T_2 and T_3 are shown with only two branch crossings.

Figure 12 Phylogenetic trees on *FoxP2* gene DNA sequences of 11 species, (a) phylogenetic trees T_1 and T_3 in (12) (b) phylogenetic trees in rearranged taxa order: T_1 vs. T_3 (c) phylogenetic trees in rearranged taxa order: T_2 vs. T_3



6 Conclusions

This article suggests visualising two alternative phylogenetic trees together on a fixed taxa order. To do so requires solving the branch crossing minimisation problem. A naïve algorithm to find the minimum number of branch crossings was introduced and an efficient algorithm to count the number of branch crossing in an α-centric dendrogram. More efficient algorithms or further studies are needed to solve the branch crossing minimisation problem.

The phenogram was emphasised in this article and the branch crossing minimisation in cladogram is another future work. Other future work includes visualising a pair of dendrograms of two different taxa sets as in Figure 11.

References

Amenta, N. and Klingner, J. (2002) 'Case study: visualizing sets of evolutionary trees', *Proceedings of IEEE Information Visualization*, Boston, MA, pp.71–74.

Bar-Joseph, Z., Gifford, D. and Jaakkola, T. (2001) 'Fast optimal leaf ordering for hierarchical clustering', *Bioinformatics*, Vol. 17, No. Suppl. 1, pp.S22–S29.

Cha, S-H. (2013) 'On visualizing a pair of phylogenetic trees', *Proceedings of Mathematics and Computers in Biology and Biomedical Informatics*, Baltimore, MD, pp.73–76.

Duda, R.O., Hart, D.G. and Stork, D.G. (2000) *Pattern Classification*, 2nd ed., Wiley, New York.

Dunn, G. and Everitt, B.S. (1982) *An Introduction to Mathematical Taxonomy*, Cambridge Dover Publications, Inc., Mineola, New York.

Dwyer, T. and Schreiber, F. (2004) 'Optimal leaf ordering for two and a half dimensional phylogenetic tree visualisation', *Proceedings of the Australasian Symposium on Information Visualization*, Christchurch, New Zealand, pp.109–115.

Goldberg, D.L. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.

Michener, C-D. (1970) 'Diverse approaches to systematics', *Evolutionary Biology*, Vol. 4, pp.1–38.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms*, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Nye, T.M.W., Lio, P. and Gilks, W.R. (2005) 'A novel algorithm and web-based tool for comparing two alternative phylogenetic trees', *Bioinformatics*, Vol. 22, No. 1, pp.117–119.

Robinson, D-F. and Foulds, L-R. (1981) 'Comparison of phylogenetic trees', *Mathematical Biosciences*, Vol. 53, Nos. 1–2, pp.131–147.

Scornavacca, C., Zickmann, F. and Huson, D-H. (2011) 'Tanglegrams for rooted phylogenetic trees and networks', *Bioinformatics*, Vol. 27, No. 1, pp.248–256.

Scornavacca, C., Zickmann, F., Huson, D-H., Buchin, K., Buchin, M., Byrka, J., Nöllenburg, M., Okamoto, Y., Silveira, R. and Wolff, A. (2012) 'Drawing (complete) binary tanglegrams', *Algorithmica*, Vol. 62, Nos. 1–2, pp.309–332.

Sokal, R.R. and Rohlf, F.J. (1962) 'The comparison of dendrograms by objective methods', *Taxon*, Vol. 11, No. 2, pp.33–40.

Woese, C-R., Gibson, J. and Fox, G.E. (1980) 'Do genealogical patterns in purple photosynthetic bacteria reflect interspecific gene transfer?', *Nature*, 10 January, Vol. 283, No. 5743, pp.212–214.

Zainon, W.N.W. and Calder, P. (2006) 'Visualising phylogenetic trees', *Proceedings of the 7th Australasian User Interface Conference*, Vol. 50, pp.145–152.