A Classification Based Approach for Root Unknown Phylogenetic Networks under Constrained Recombination

M. A. H. Zahid[1], Ankush Mittal[1], R. C. Joshi[1]

[1] Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee Uttaranchal, India 247667.
{Zaheddec, ankumfec, joshfec}@iitr.ernet.in

Abstract. Phylogenetic networks are the generalization of the tree models used to represent evolutionary relationship between the species. Tree models of evolutionary process are not adequate to represent the evolutionary events such as, hybridization, lateral/ horizontal gene transfer and genetic recombination. A well-formulated problem in phylogenetic networks, due to recombination, is to derive a set of input sequences on a network with minimum number of recombinations. No efficient algorithm exists for this problem as it is known to be NP-hard. Efficient solutions exist for the constrained recombination networks, where the nodes on each recombination cycles are disjoint. These solutions are based on the assumption that the ancestral sequence is known in advance. On the other hand, the more biologically realistic case is that where the ancestor sequence is not known in advance. In this paper we propose an efficient classification based method for deriving a phylogenetic network under constrained recombination without knowing the ancestral sequence.

1 Introduction

The phylogenetic tree construction methods failed to find true relationship between the species, because of the evolutionary events such as, horizontal gene transfer, hybridization, homoplasy and genetic recombination. The network representation of the evolutionary relationship provides a better understanding of the evolutionary process and the non-tree like events [1, 2]. Detection of recombination plays in important role in locating the origin of the gene influencing the genetic disease. A case study on HIV, carried at the center for computational and experimental genomics, Department of Biological Sciences, University of Southern California shown that the most frequent recombination event make it difficult to design a drug for HIV. Recombination in HIV is recognized as an important mechanism by which the viruses escape the attack against the drug [3].

Since long time, the consequences of the recombination are ignored, and phylogenies were constructed by neglecting the recombination events. Schierup and Hein in [2, 6] and Posada [7] shown the effect of negligence of recombination while con-

structing the phylogeny. When recombination occur different parts of the genetic sequence represents different histories violating the conventional assumption of a sequence representing single underlying history. Despite of this fact, very little have been published on the methods robust for recombination. A good amount of work has been done for non tree like evolutionary events other than recombination, for an exhaustive survey refer [8, 9].

Wang et al. [4] showed the problem of finding a perfect phylogenetic network, network with minimum number of recombination nodes, is NP-hard and gave an algorithm for a restricted problem, called node disjoint network, with $O(n^4)$ computing time. The restriction is that in a merge path of a recombination node, there is no node that is in the merged path of a different recombination node. In other words, no node is shared by two or more recombination cycles, also called as "gall". The phylogenetic network, in which every recombination cycle is a gall, is also called a "gall tree". The network construction methods in [4, 5] construct the phylogenetic network with the assumption that the ancestral sequence is known in advance. On the other hand more biologically realistic case is that the ancestral sequence is not known in advance. Gusfield [12] proposed an algorithm similar to [5, 11] for unknown root and used the concepts of split graphs and conflict graphs to construct the phylogenetic network. The algorithm given in [12] computes the root unknown gall tree in $O(nm + n^3)$ time. In this paper we proposed a classification technique, based on biological constraints, for the classification of the nodes in the network. Theses classified nodes are used to construct the phylogenetic network for unknown ancestor sequence. The nodes are classified into mutation, recombination and null classes. The proposed method takes $O(n\log n + mn^2)$ computing time for classifying all the nodes.

The paper is organized as follows. Section 2 deals with the formal definitions and assumptions related to phylogenetic networks. Section 3 deals with the combinatorial background and conditions for the detection of the recombination. The algorithm for classifying the nodes with an example is given in section 4.

2 Preliminaries

This section deal with the basic terminology and assumptions made for the development of algorithm. We followed the terminology from [5] and [11] for simplicity.

Formally, a phylogenetic network is a directed acyclic graph, but underlying undirected graph can have cycles. Each node in the phylogenetic network *N* has indegree 0, 1 or 2. The nodes with indegree 0 are called independent node as the ancestor to these nodes is unknown, the nodes with indegree 1 are called tree nodes and the nodes with indegree 2 are called recombination nodes. A tree node is the result of mutation and the recombination node is the due the recombination of genetic material of two parent species of the node. Each node in the network *N* is assigned a binary sequence of length *m*. The tree or mutation nodes have a single site or character change from 0 to 1, when compared with the parent nodes. The sequence of recombination node is the parts of its two ancestor's sequences.

If a node *u* is reachable from a node *v* via a directed path, then *v* is an ancestor of *u*, and *u* is the descendent of the node *v*. Each node in the phylogenetic network is represented with a binary number of some specified length *m*. In the perfect phylogeny the transformation of states from 0 to 1, occurs at most ones for each site or the column in the binary sequence. The nodes on perfect phylogenetic networks are organized in such a way that there is unique node having state 1 in site $i, i \in m,$ every other node having 1 at site *i* is the descendents of this unique node. The transformation from 0 to 1 is possible in case of recombination, where the crossovers can change the state from 0 to 1. A phylogenetic network with recombination is said to be perfect if it has minimum number of recombination nodes and follows all the restrictions mentioned above.

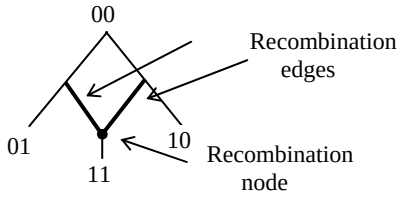

Fig. 1. Phylogenetic network for binary sequences

A set of binary sequences represents a phylogenetic network *N*, if and only if each sequence labels exactly one leaf of the network *N*. A phylogenetic network on a set of three binary sequences is shown in Fig. 1. The biological interpretation of a phylogenetic network *N*, for *M* binary sequences is that the network represents the possible history of the *M* sequences under the following assumptions. (1) The change in one site, from 0 to 1, is permitted only once (called mutation). (2) Two sequences are permitted to recombine as a result of recombination event. (3) Each site in the sequence represents a SNP (single nucleotide polymorphism), a site where two of the four possible nucleotides appear in the population with the frequency above some threshold [10].

Given a set of species *n* species with binary sequences of length *m*, a phylogenetic network with $O(nm)$ recombination nodes exist. Recombination is a rare event in the evolutionary process. Therefore a phylogenetic network with minimum number of recombination nodes is informative.

3 Conditions for the classification of nodes

In this section we formulate the necessary and sufficient condition for the classification of nodes, which has biological significance. We used the similarity and dissimilarly between the sequences as the major tool for the classification of the nodes into mutation, recombination and null classes.

Lemma 1 is crucial for the detection of the recombination cycles in the given binary sequences. It states that the similarity and dissimilarity between the sequences, which shares a common parent, should be computed after the removing the parent's

characteristics are removed from each of the child, to avoid the misleading similarity between the species.

Lemma 1. *Let S and S' be the sequences of the children of node v. if S' is not the result of the mutation or recombination in S then the similarity between S and S' is due to common ancestry.*

Proof. Let *S'* is not a child of the *S*, then *S'* is not reachable from *S*, therefore all the sites or character of *S'* are different from the characters of the node *S* or vise versa. Let *S* and *S'* are children of node *v*, then according to the assumption made in section 2 both *S* and *S'* are reachable from node v, and show the similarity by at least one character (of site) with the parent node *v*. Both the nodes *S* and *S'* show the similarity with their parent node by at least one character not with each other. Hence this proves that the distinct nodes will show similarity due to common ancestry. □

Lemma 2 gives a method of finding child node and parent node when the compared nodes show some similarity.

Lemma 2. If a node *v'* is the result of mutation from its parent *v* then $v < v'$, when the sequences are considered as the binary numbers.

Proof. We prove this by mathematical induction on the length *m* of the binary sequences. In the first step consider a parent *v*, with sequence *S*, contains all 0's in its sequence. According to the definition of the mutation only one site can change the state from 0 to 1 and rest of the sequence remains same. If a mutation occurs at site *i* of *v* leading to at least on of the sites of the sequence, *S'*, of the node *v'* is set to 1 and the rest of the sequence will remain same as the parent sequence. Thus making *S* less than *S'*, *S<S'*. Now consider the case where the node *v* has *m-2* number of 1s in sequence *S*. A mutation leads to *m-1* number of ones in *S'* making *S<S'*. Now we prove it for a generalized case of *m-1* number of 1s. If a node *v*, with sequence *S* having *m-1* number of 1s mutates to result in new child node *v'* with sequence *S'* having *m* number of ones, which is the highest value binary number for a given length *m*. therefore $v < v'$ when mutation is reason for speciation. □

Lemma 3 plays an important role in the detection of the recombination nodes. It proves that if a node is the result of recombination then it should be greater than at least one of the parents.

Lemma 3. Let *v* be a recombination node with sequence *S*. if *P'* and *P''* are two parent nodes of *v*, with sequences *S'* and *S''* respectively, then any of the following should hold.

(a) S'>S and S''>S
(b) S'>S and S''<S
(c) S'<S and S''>S

Proof. To prove this it is enough to prove it for the binary sequences of length 2. Let three binary sequences, which give a recombination node *v* are 00, 01, 10, 11. These sequences can be placed in only three different ways to represent the recombination as shown in Fig.2. The other possibilities are ruled out due to the assumption that back mutation is not permitted.

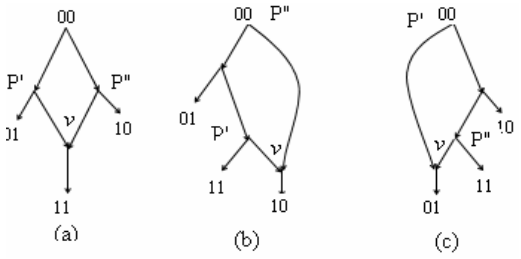

Fig. 2. Three cases for lemma 3

Case (a): Here the two mutations from root node lead to the species 01 and 10. The node *v* with sequence 11 is the result of recombination of 01 and 10. Clear *v* is greater than its two parents.

Case (b): In this case the sequence 01 muted from root and the sequence 11 is mutated from 01. The recombination node *v* is the result of recombination between root and *P'*. It satisfies the case (b) stating, *S'>S* and *S"<S*.

Case (C): In this case the sequence 10 muted from root and the sequence 11 is mutated from 10. The recombination node v is the result of recombination between root and *P"*. It satisfies the case (c) stating, *S'<S and S">S*. □

Theorem 4 uses the lemma 2 and 3 for the detection of the recombination nodes. It helps in finding the parents of the recombination nodes when any one of the parent is greater than the child.

Theorem 4. Let *M* be the given sequence matrix representing the node disjoint network. A species or sequence is said to the result of recombination if it holds any of the following conditions.

(a) If two species have, *0<similarity<=100%* and *dissimilarity > (100/m) %*, where m is the length of the sequence, one sequence represent parent and another sequence represent the child, which is the result of recombination.
(b) The similarity between the two parent of a recombination node is always 0.

Proof. Case (a): Suppose that at some node *x*, mutation occurred at site *i*, representing the change at site *i* from 0 to 1 and rest of the sequence remain same. If we calculate its similarity and dissimilarity corresponding to the value 1 at each site, the similarity will be *100%* and dissimilarity will be *100/m %* exactly. This indicates that only one site has modified its value from 0 to 1. By the assumptions we made for phylogenetic network, there is no provision for back mutation, that is transformation from 1 to 0 or mutation of more than one site at the same instance of time is also ruled out. This restricts the dissimilarity to be exactly *100/m %* for mutation. But in case of recombination the restrictions of the mutation are ruled out due to the fact that the resulting sequence may carry a part of the sequence from one parent and rest will be imitated from the other parent (in single crossover). This fact indicates that the similarity can be *0<similarity<=100%* and *dissimilarity > (100/m) %*. Hence the condition (a) is proved.

Case (b): We prove this by contradiction. Suppose the recombination node *v* has two parents with the sequences *S'* and *S"*, show some similarity with each other. From the assumptions made in section 2 and lemma 1, the similarity between the species is due

to two reasons (1) common parent, (2) child parent relationship with a single mutation and (3) due to recombination. If *S'* and *S"* shows some similarity then any of the above relation holds. The relation 1 is avoided by removing the parent characteristics while computing the similarity between the children. We are focusing on a constrained recombination problem, where two recombination nodes are disjoint, avoid the relation 3. If *S'* and *S"* shares child parent relationship with mutation then the result of recombination will be a sequence N, similar to child or parent instead of the new sequence, $N \in (S', S'')$. Hence it's proved that the parents of the recombination node are dissimilar to each other. □

Theorem 5 gives a strong basis for the detecting the node disjoint network in the given input data. Any data satisfying the conditions given in theorem 5 will have a gall tree. Otherwise, the data does not represent the gall tree structure.

Theorem 5. If C, C', and C" are child list of sequences S, S', and S" then the following conditions should hold for the gall trees.

(a) If $C \cap C' \neq \phi$ and $|C \cap C'| = 1$.

(b) If the number of recombinations node in any of the parents is greater than 1, and $C \cap C' \neq \phi$ then $C \cap C'' = \phi$ or $S'' \cup C'' \subseteq C$ and $C' \cap C'' = \phi$ or $S'' \cup C'' \subseteq C'$.

Proof. Case (a)**:** This is proved by contradiction. Let $|C \cap C'| > 1$, represents that the node *S* and *S'* are involved in more than one recombination with each other. The path from root node to the recombination node always has two alternatives, each from one of its parents. If there are more than two recombination nodes for the single pair of parents *S* and *S'*, then there are two paths for each recombination nodes which involves the same set of parents *S* and *S'*. In other words the parent nodes are shared by two recombination cycles. But according to the definition of node disjoint network, the nodes on the path to one recombination node should not be shared with other recombination node path Hence this rejects our hypothesis and proves the condition.

Case (b): The proof is similar as in case (a). Let $C \cap C' \neq \phi$ and the number of recombination nodes in C are two. If $C \cap C' = x$ and $C \cap C'' = y$, then there exist a path from root node to the recombination cycle of node *x* and node *y*, which passes through the node C. This violates the node disjoint rule of phylogenetic networks. Let child list *C"* and the node *S"* itself is a subset of the child list of node *S.* If the similarity and dissimilarity between the children of *S* is computed after removing the *S*'s characteristics from each node then the recombination node will not show *S* in its parent list, according to lemma 1. This parent relationship with the recombination node is due to the common ancestor of all the nodes in the recombination cycle. So when there is common ancestor for the parents of a recombination node then the parent of all the nodes in that cycle is also added as the parent to the recombination node. □

4 Phylogenetic Network Reconstruction Algorithm

In this section we develop a formal algorithm for the phylogenetic network reconstruction with constrained recombination and prove that this algorithm results in

minimum number of recombinations in the resulting network. We conclude the section with an example for the algorithm.

4.1 The *Node_Class* Algorithm

The algorithm *Node_Class* classifies the nodes and make the child and parent list of each node given in the data matrix based on similarity and dissimilarity. We assume that all the sequences represent a unique leaf node in the network, and back mutation is not permitted.

The algorithm accepts a $n \times m$ binary matrix as input, where each row represents a node in the phylogenetic network. A similarity and dissimilarity matrix is generated based on the input matrix and is computed corresponding to the value 1 at the sites. The distance (similarity and dissimilarity) between the siblings is measured after removing the parent's characteristics from the children. The parent node is considered as the root to all the nodes in the child list and the parent list represents the parent nodes of the current node. If data does not represent node disjoint network the algorithm terminates by reporting an error message. The algorithm is as follows.

Data structures:
$d \leftarrow$ *is an input matrix of size* $n \times m$ *,where n is number of species and m is length of sequences.*
$sim\_dis_{ij} \leftarrow$ *is the similarity and dissimilarity matrix corresponding to 1's in the sequences.*
Node is a record with three variables: Label, Count (number of parents), and Type(class).
$Child_i \leftarrow$ *An array of child nodes labels for each node.*
$Parent_i \leftarrow$ *An array of parent nodes labels for each node.*
INPUT: - binary matrix of $n \times m$ *size.*
OUTPUT: - child list for each node.

ALGORITHM: *Node_Class* (*d*)
Sort the matrix by considering each row as binary number.
for each row in the input binary matrix **do**
Label $\leftarrow$ *row_value*;
Count $\leftarrow$ 0;
Type $\leftarrow$ *Null*;
for each sorted node $1 \le i \le n$ **do**
$Child_i \leftarrow$ *Null*;
$Parent_i \leftarrow$ *Null*;
for each node $1 \le j \le n$ **do**
if $sim\_dis_{ij} \leftarrow$ *Null* **then**

Compute *Similarity* and *dissimilarity* between *i* and *j*;
Modify sim_dis matrix;
endif;
if $Node_i.Type = Null / mutation$ **then**
if *Similarity = 100%* and *Dissimilarity = 100/m %* **then**
$Node_j.Count \leftarrow Node_j.Count + 1;$
$Parent_j \leftarrow Parent_j \cup Node_i;$
else if $Node_j.Count \leq 2$ **then**
$Noide_j.Type = recombination;$
$Child_i \leftarrow Child_i \cup Node_j;$
endif;
else
$Noide_j.Type = mutation;$
$Child_i \leftarrow Child_i \cup Node_j;$
endif;
if *Similarity < 100% and Dissimilarity* $\geq$ *100/n %* **then**
$Node_j.Count \leftarrow Node_j.Count + 1;$
$Parent_j \leftarrow Parent_j \cup Node_i;$
$Noide_j.Type = recombination;$
$Child_i \leftarrow Child_i \cup Node_j;$
endfor;
for each x, $1 \leq x \leq |Child_i|$, and $Node_x \in Child_i$ **do**
Compare $Node_i$ with other element of $Child_i$ after removing parents characteristics;
Modify sim_dis matrix;
endfor;
endfor;
Test_Nodedis (*child, Node*)
return;
endAlgorithm;

The function *Test_Nodedis*, takes Child and Node record list as input and based on theorem 5 verifies whether node disjoint network exist in the given data or not. The function is as follows.

Function: *Test_Nodedis* (*Child, Node*)
for each node $1 \leq i \leq n$ **do**

if number of recombination nodes > 1 or $Node_i.Count > 2$ **then**
  for each node $1 \le j \le n$, where $j \ne i$ **do**
    if $|Child_i \cap Child_j| > 1$ **then**
      exit "node disjoint recombination cycle does not exist";
    else
      for each node $1 \le k \le n$, where $k \ne i, j$ **do**
        $test \leftarrow Node_k \cup Child_k$;
        if $Child_i \cap Child_k \ne \phi$ or $Test \not\subset Child_i$ and
          $Child_j \cap Child_k \ne \phi$ or $Test \not\subset Child_j$ **then**
          exit "node disjoint recombination cycle does not exist";
        else
          Compute the similarity and dissimilarity matrix, *d*, after removing the parent's characteristics from each child;
          Node_Class(*d*)
      endif;
    endfor;
  endif;
endfor;
return "node disjoint recombination cycle exist";
endFunction;

4.2 An Example

The input matrix for the algorithm is shown in Fig. 3(a), which consists of seven leaf nodes with their binary sequences. As the first step in the algorithm we sort the nodes considering each row represents a node and is a binary number. The sorted matrix is shown in Fig. 3(b).

A	0	0	0	1	0
B	1	0	0	1	0
C	0	0	1	0	0
D	1	0	1	0	0
E	0	1	1	0	0
F	0	1	1	0	1
G	0	0	1	0	1

(a)

A	0	0	0	1	0
C	0	0	1	0	0
G	0	0	1	0	1
E	0	1	1	0	0
F	0	1	1	0	1
B	1	0	0	1	0
D	1	0	1	0	0

(b)

Fig 3. (a) Input binary matrix with labels.
(b) Sorted input binary matrix on rows

After processing each node the values assigned to each variable or properties of the node records is shown in Table 1. The Type values for nodes A and C are 'Null' because they are mutated from the root node, not from any given nodes. The nodes D and F are the result of the recombination and have two parents. All the other nodes are the result of mutation from their respective parents.

Table 1. Values of each property of node record after the processing input matrix shown in Fig. 3(a)

Node Label	Type	Count
A	Null	0
B	Mutation	1
C	Null	0
D	Recombination	2
E	Mutation	1
F	Recombination	3
G	Mutation	1

	A	B	C	D	E	F	G
A	1,0	1,1	0,2	0,3	0,3	0,4	0,3
B	1,1	2,0	0,3	1,2	0,4	0,5	0,4
C	0,2	0,3	1,0	1,1	1,1	1,2	1,1
D	0,3	1,2	1,1	2,0	0,2	0,3	0,2
E	0,3	0,4	1,1	0,2	2,0	1,1	0,2
F	0,4	0,5	1,2	0,3	1,1	3,0	1,1
G	0,3	0,4	1,1	0,2	0,2	1,1	2,0

Fig 4. Similarity and dissimilarity matrix for the input data shown in Fig. 3 (a)

The similarly-dissimilarity matrix computed during the detection of the recombination nodes is shown in Fig. 4. The node C is parent for the nodes D, E, F, and G, so the similarity dissimilarity measure between the children is computed after removing the parent's characteristics.

Table 2 shows the child list of each node. The nodes D and F don't have any child so their list carries *Null* entry. On the other hand the nodes D and F are in the child list of (B, C) and (E, G) nodes respectively, making D and F, recombination nodes. Table 3 gives the list of parent nodes for each node. This list is computed based on the node disjoint conditions proved in theorem 5. The child list for the node C have two recombination nodes, D and F, and the child F has count value 3 indicating three parents. But it satisfies second condition in theorem 5, therefore the count is reduced by 1 and its super ancestor is removed from its parent list.

Table 2. Child list for the input matrix shown in Fig. 3(a).

Node Label	Child List
A	B
B	D
C	D,E,F,G
D	Null
E	F
F	Null
G	F

Table 3. Parent list for the input matrix shown in Fig. 3(a)

Node Label	Child List
A	Null
B	A
C	Null
D	B,C
E	C
F	E,G
G	C

Given the child and parent list for each of the node in the input data, it is easy to construct the phylogenetic network for it. The procedure starts with scanning the child list table. For each node in the child list a cross verification is performed with the parent list. If both validate each other, then nodes are added and connected accordingly in the child parent relationship. Otherwise, an additional node is created which has the same sequence as conflicting node, called the coalescent node, and

child and parent tables are modified. Each internal node is attached with a new node representing the leaf node in the node disjoint network, except the coalescent nodes. The sequence for the new node is same as its parent. The final network for the input data given in Fig. 3(a) is shown in Fig. 5.

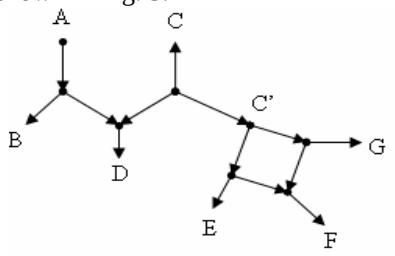

Fig. 5. Node disjoint network for the input shown in Fig. 3(a)

Now we prove that the algorithm results in a node disjoint network, if one exists, with the minimum number of node disjoint recombination cycles in it.

Theorem 6. *If the for the input matrix M there are k recombination nodes then any node disjoint network that minimizes the recombination will have exactly k recombinations.*

Proof. Let *T* be a node disjoint network for the input binary matrix *M*. If there is a node disjoint cycle *Q* in *T* that contains only the mutation nodes, then the sequence labeling of the nodes on *Q* can be derived from the perfect phylogeny. The root of the node disjoint cycle *Q* is the sequence labeling the coalescent node of *Q*. Replacing *Q* with perfect phylogeny will result in a node disjoint network with one recombination less than the network *T*. Hence in any node disjoint network using the minimum number of recombinations must have exactly one recombination node for each node disjoint cycle. Therefore the minimum number of node disjoint cycles in a node disjoint network is exactly the number of nodes with recombination type or class.

4.3 Correctness and time complexity

The results in section 3 and 4 give the proof of correctness of the classification method. When the input data does not display a node disjoint network structure, the algorithm reports an error message and terminates. The phylogenetic network computed by the algorithm will have minimum number of recombinations.

The algorithm computes a node disjoint network, if one exists, in $O(n\log n + n^2 m)$ time, where *n* is number of nodes, and *m* is the length of the each binary number. The algorithm sorts the *n* rows, considering each row as a binary number, using quick sort, which takes $O(n\log n)$ time. In the next step the algorithm computes the similarity and dissimilarity between each of the node with respect to the sites with the value 1. The second step takes $O(n^2 m)$ time. On the basis of the similarity and dissimilarity measure the type of each node is decided, and at the end of every iteration the child

and parent list for a specific node is created. This child list can be further used to construct the node disjoint network.

5 Conclusions

In this paper we proposed a classification based approach for the construction of phylogenetic network with constrained recombination for unknown root or ancestor. The construction of perfect phylogenetic network is proved to be NP-hard by Wang et al. [4]. Wang et al. [4] gave a polynomial time algorithm for a restricted problem called node disjoint network with known root, in which a node can not be a part of two recombination paths in the network. It has both algorithmic and biological significance. The method in [4] computes the gall tree or node disjoint network in $O(n^4)$ time. Guesfield et al. [5, 11] proved that the [4] does not give the necessary and sufficient conditions for the gall tree construction and gave a sufficient combinatorial basis for network construction with known root. A similar method as [5, 11] is given by Guesfield et al. [12] for the construction of the node disjoint network for unknown root. The method [12] takes $O(nm + n^3)$ time for constructing a network for unknown root.

The proposed algorithm computes the root unknown network in $O(n \log n + n^2 m)$ time and established the necessary and sufficient condition for the root unknown networks. Unlike the other algorithms, we followed a row-based search to detect the recombination nodes. Other algorithms search the columns for the detection of recombination. The number of columns in a sequence may be far greater than the row, which increases the complexity of the previous algorithms.

References

1. Posada, D., Crandall, K.: Intraspecific gene genealogies: trees grafting into networks, Trends in Ecology and Evolution. 16 (2001) 37–45.
2. Schierup, M. H., Hein, J.: Consequences of recombination on traditional phylogenetic analysis. Genetics. 156(2000) 879-891.
3. Savai, P., Abulleef, H., Chun, L. L., Skvortsov, D.: Phylogenetic analysis, MS. Project, University of southern California, 2002.
4. Wang, L., Zhang, K., Zhang, L.: Perfect phylogenetic networks with recombination. Journal of Computational Biology. 8 (2001) 69-78.
5. **Guesfield**, D., Satish, E., Langley, C.: Optimal efficient reconstruction of phylogenetic network with constrained recombination. Journal of Computer and System science. 70 (2005) 381-398.
6. Schierup, M. H., Hein, J.: Recombination and the molecular clock. Mol. Biol. Evol. 17(2000) 1578–1579.
7. Posada, D., Crandall, K.: The effect of recombination on the accuracy of phylogeny estimation. Journal of Molecular Evolution. 54(2002) 396-402.
8. Linder C.R., Moret, B.M.E. L. Nakhleh, and T. warnow, Reconstructing networks part II: computational aspects. A tutorial presented at the ninth pacific symposium on Biocomputing (PSB), 2004.
9. Zaidi, M. A. H., Mittal, A., Joshi, R. C.: Use of phylogenetic networks and its reconstruction algorithms. Journal of Bioinformatics India, ISSN 0972-7655. 4(2004) 47-58.

10. Chakravarthi, A., It's raining SNP's hallelujah? Nature Genetics. 19 (1998) 216-866.
11. Guesfield, D., Satish, E., Langley, C.: The fine structure of galls in phylogenetic networks. INFORMS J. on computing, special issue on Computational Biology. 16(2004) 459-469.
12. Gusfield, D.: Optimal, Efficient Reconstruction of Root-Unknown Phylogenetic Networks with Constrained and Structured Recombination, J. Computer and Systems Sciences, Special issue on Computational Biology, 70 (2005) p. 381-398.