RC23837 (W0512-116) December 23, 2005
Computer Science

IBM Research Report

A PQ Tree-based Framework for Reconstructing Common Ancestor under Inversions and Translocations

Laxmi Parida
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

A PQ Tree-based Framework for Reconstructing Common Ancestors Under Inversions & Translocations

Laxmi Parida

Computational Biology Center, IBM TJ Watson Research Center,
Yorktown Heights, New York 10598, USA
parida@us.ibm.com

Abstract. Various international efforts are underway to catalog the genomic similarities and variations in the human population. Some key discoveries such as inversions and translocations within the members of the species have been made in the last few years. The task of constructing a correct genealogy tree of the members of the same species, given this knowledge and data, is an important problem. In this context, a key observation is that the "distance" between two members, or member and ancestor, within the species is small. In this paper, we pose the tree (genealogy) reconstruction problem exploiting some of these peculiarities. Our proposed scheme is based on the idea of minimal consensus PQ trees on permutations. We give a simple scheme to construct the tree and use a modified PQ tree structure to reconstruct the common ancestors on the genealogy tree.

Keywords: PQ tree, inversion, reversal, translocation, genome rearrangement, phylogeny, genealogy, common ancestor, tree construction.

1 Introduction

Various international efforts are underway to catalog the genomic similarities and variations in the human population [7, 5, 3, 36]. As the study progresses, data in the form of genomic markers is becoming available, with due respect to individuals' and groups' privacy, for public study and use. Combined with recent discoveries of inversion and translocation within the human species, this opens up the potential for using large scale rearrangements to reconstruct the genealogy tree of the human population.

We first give a brief summary of discovered inversions and translocations within the human population and then briefly review the computational methods being used by the bioinformatics community to tackle the problem of reconstructing phylogeny trees.

Inversions along a chromosome is frequently observed by comparing closely related species: for example, chimpanzee chromosome 19 and human chromosome 17 [14], mouse chromosome 16 and human chromosome 21 [19]. These are generally very long inversions that are observed as reversed gene orders [25]. However inversions have been seen across humans: X chromosome [34] and a 3 Mb inversion on the short arm of the Y chromosome [12]. Human inversions occur at a low but detectable frequency. The ones that are large enough to be detected by standard cytogenetic analysis occur at a frequency of 1-5 per 10,000 individuals [17]. The inversions across humans are of particular interest, since often the recombination in the inverted segments in heterozygotes lead to heritable disorders [18, 13].

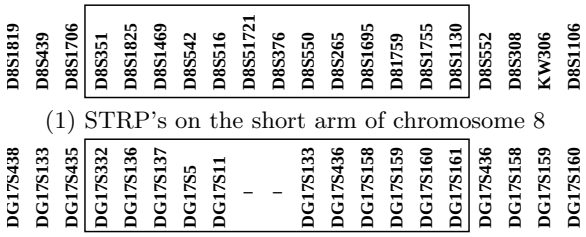
Secondly, inversions also have a potential for explaining the geographic distribution of the human population: a reconstruction of the prehistoric human colonization of the planet [5, 3, 36]. The X-chromosome inversion is seen in populations of European descent at a frequency of about 18% [34]. Further large chromosomal segment inversion have been seen in humans: [20] reports a paracentric[1] inversion polymorphism spanning larger than 2.5 Mb segment in chromosome band $8p23.1 - 8p22$ and [15] reports a 900-Kb inversion on chromosome 17q21-31. The second inversion is seen at the rate of 20% in Europeans and almost absent in East Asians and rare in Africans.

Large chromosomal rearrangement polymorphisms such as deletions or duplications is apparent by loss or gain of heterezygosity. However inversions are difficult to detect and may go unnoticed if the inverted segment is small.

[1] An inversion not involving the centromere.

The inversions may occur in coding, non-coding, or intra-gene regions of the chromosome. Hence a model that tracks the gene orders of the chromosome is inadequate for modeling segment inversions. Instead, these inversions [20, 15] are being discovered and reported in terms of the order of the labeled STRP's (Short Tandem Repeat Polymorphisms). See Figure 1 for two instances of inversions in the human chromosomes. Further, unlike genes, these markers are not signed (say, as used in [4]). Also the ancestral segment is unknown, i.e. it is unclear which order of the segment came first.

Translocations have also been observed in humans [10, 11], although these have been mostly of single genes and generally associated with a disorder. It is believed that as we progressively learn about individual differences, more such variations, translocations or inversions, will surface. In fact according to [15], *these* (inversions) *may be only the tip of the iceberg.*



(1) STRP's on the short arm of chromosome 8

(2) STRP's on the long arm of chromosome 17

Fig. 1. The Short Tandem Repeat Polymorphisms on two human chromosomal segments. The blocked segment shown here is inverted in a significant fraction of the human population.

Computational Background. Loosely speaking there are two computational approaches to studying the evolutionary relationships of genomes, one of studying the individual gene sequences and the other of studying the arrangement of multiple genes on the genome. A very large amount of literature exists for the first approach (including sequences under the character model), which we will not discuss here to avoid digression. The second approach was initiated by Sankoff [31]: the description of chromosomal inversions in Drosophila had appeared way back in 1938 [8] and there also had been other strong evidences in favor of whole genome rearrangements [30, 29]. An active interest has been taken in the study of genome rearrangements in the last decade resulting in some very interesting observations and debates in the community.

In the context of genome rearrangements, genomes are viewed as permutations where each integer corresponds to a unique gene or marker. For monochromosomal genomes, the most common rearrangement is *inversion* that is often called *reversal* in the area of bioinformatics. Without loss of generality a permutation of length n with $i \leq j$, can be written as π_1, the inversion on π_1 defined as $r^{ij}(\pi_1)$ and the translocation on π_1 defined as $t^{ijk}(\pi_1)$ where the boxed portion is the reversed or translocated segment.

$$\begin{aligned}
\pi_1 &= p_1 p_2 \ldots p_{i-2} p_{i-1} \boxed{p_i p_{i+1} p_{i+2} \cdots p_j} p_{j+1} p_{j+2} \cdots p_k p_{k+1} \cdots p_n \\
r^{ij}(\pi_1) &= p_1 p_2 \ldots p_{i-2} p_{i-1} \boxed{p_j p_{j-1} p_{j-2} \cdots p_i} p_{j+1} p_{j+2} \cdots p_k p_{k+1} \cdots p_n \\
t^{ijk}(\pi_1) &= p_1 p_2 \ldots p_{i-2} p_{i-1} p_{j+1} p_{j+2} \cdots p_k \boxed{p_i p_{i+1} p_{i+2} \cdots p_j} p_{k+1} \cdots p_n
\end{aligned}$$

Clearly, $r^{ij}(r^{ji}(\pi)) = \pi$ leading to the idea of a shortest inversion distance between two permutations. Let the shortest inversion distance between π_1 and π_2 be given as $D^r(\pi_1, \pi_2)$. However, computing $D^r(\pi_1, \pi_2)$ for a a given pair of permutations π_1 and π_2 is NP-complete [6]. Hannanhelli and Pevzner showed that by supplementing the genes with signs, this problem could be solved in polynomial time by using graph structures termed *hurdles* and *fortresses* [23]: this is perhaps the most cited work in the area of computational genome rearrangements. The central idea has been subsequently conceptually simplified by Bergeron and Stoye [1, 2].

In sequences, the problems of (1) multiple sequence alignment and (2) the construction of the implicit phylogeny tree, have been traditionally separated for obvious reasons [22, 21, 35]. Such a distinction under the genome rearrangement model is not so obvious. However breakpoint phylogeny was introduced by Sankoff and Blanchette [32] to study this problem under a simplified cost function of minimizing the number of breakpoints. Heuristic approaches also have been applied to this problem in [33, 4]. A rich body of literature

on inferring phylogenies under the sequence or character models exists including attempts at using sequence and distance based methods to genome rearrangement problems [16, ?].

Contributions of this paper. In this paper we present a very simple computational model of the multiple genome rearrangement problem. Since the motivation is from ordered chromosomal segments, we deal with *unsigned permutations*. Further, since the inversions and translocations are within the same species, the distance between the members is observed to be very small. The *coalescent* approach, which focuses mainly on mutations at a fixed site, is based on the realization that genealogy is usually easier to model backward in time [24]. We take a similar approach to the large scale genome rearrangements model.

The central idea of the paper is based on the notion of minimal consensus PQ tree T of permutations introduced in [28] and the observation that the number and size of each (excluding leaf nodes) is $O(1)$ for a small distance between permutations. We also propose an annotation scheme (called oriented PQ tree), that helps uniquely reconstruct the permutations from the tree. Based on this we pose the problem as a permutation tree construction (PTC) task and propose a simple branch-and-bound solution. The scheme produces the genealogy tree as well as reconstructs all the common ancestors.

Roadmap. In the next section we discuss the PQ Tree data structure and the variants that we propose for the problem. In Section 3, we describe the permutation tree reconstruction problem and an efficient algorithm to compute the genealogy tree. In Section 4 we discuss the task of including mutations in the problem model and conclude in Section 5.

2 PQ Trees

This section gives a brief summary of PQ trees introduced by Booth and Leukar [26], as a tool to solve the general consecutive arrangement problem. The general consecutive arrangement problem is the following: *Given a finite set X and a collection $\mathcal{I}$ of subsets of X, does there exist a permutation π of X in which the members of each subset $I \in \mathcal{I}$ appear as a consecutive substring of π?* Booth and Leuker introduced an efficient linear time algorithm that solves this problem using a PQ tree. A PQ tree is a rooted tree whose internal nodes are of two types: P and Q. The children of a P-node occur in no particular order while those of a Q-node appear in a left to right or right to left order. We designate a P-node by a circle and a Q-node by a rectangle. The leaves of T are labeled bijectively by the elements of X.

Definition 1. ***(Equivalent PQ trees,** $T \equiv T'$**):** *Two PQ trees T and T' are equivalent, denoted $T \equiv T'$, if one can be obtained from the other by applying a sequence of the following transformation rules: (1) arbitrarily permute the children of a P-node, and (2) reverse the children of a Q-node.*

A *frontier* $F(T)$, of tree T is the sequence of leaf nodes in the left to right order. For example, in Figure 2, $F(T_1) = F(T_2) = 0123456789$. $\mathcal{C}(T)$ is defined as follows: $\mathcal{C}(T) = \{F(T')|T' \equiv T\}$.

A permutation pattern on a string is defined as a set of characters that appear possibly in different orders at different locations on the string [9]. In the same paper, a maximal notation of permutation patterns was used which was later shown to have the same structure as PQ trees. The idea of a minimal consensus PQ tree of a collection of permutations is introduced in [28]:

Definition 2. ***(minimal consensus PQ tree** $T(\Pi)$**))** *Given Π, a consensus PQ tree T of Π, written as $T(\Pi)$, is such that $\Pi \subseteq \mathcal{C}(T)$ and the consensus PQ tree is minimal when there exists no $T' \not\equiv T$ such that $\Pi \subseteq \mathcal{C}(T')$ and $|\mathcal{C}(T')| < |\mathcal{C}(T)|$.*

Of all the equivalent PQ trees in $T(\Pi)$, we are interested in some specific forms which we define below. A permutation π is *nailed* if the left to right order of π is fixed, i.e., the left uniquely refers to one end and right uniquely refers to the other end. Given Π, $T(\Pi)$ is *nailed* with respect to $\pi \in \Pi$ if the leaves ordered from left to right is the permutation π. Clearly, $T_{\pi_1}(\Pi) \equiv T_{\pi_2}(\Pi)$, for all $\pi_1, \pi_2 \in \Pi$. Next, the following convention helps us to reconstruct the two individual permutations from their nailed minimal consensus PQ tree.

Definition 3. ***(oriented PQ tree* T)** *Consider two nailed permutations π_1 and π_2 and nailed PQ tree $\boldsymbol{T}_{\pi_1}(\Pi = \{\pi_1, \pi_2\})$, without loss of generality. $\boldsymbol{T}_{\pi_1}$ is oriented if each Q node is annotated with ($\longrightarrow$) or ($\longleftarrow$) labels. The ($\longrightarrow$) label indicates that the two segments are identical in the nailed permutations π_1 and π_2. Similarly, ($\longleftarrow$) label indicates that the two segments are flipped in the nailed permutations π_1 and π_2. Further, a P node with k children is numbered by integers 1 to k denoting the order in which they appear in π_2 (they appear in the left to right order in π_1 as depicted in the oriented PQ tree).*

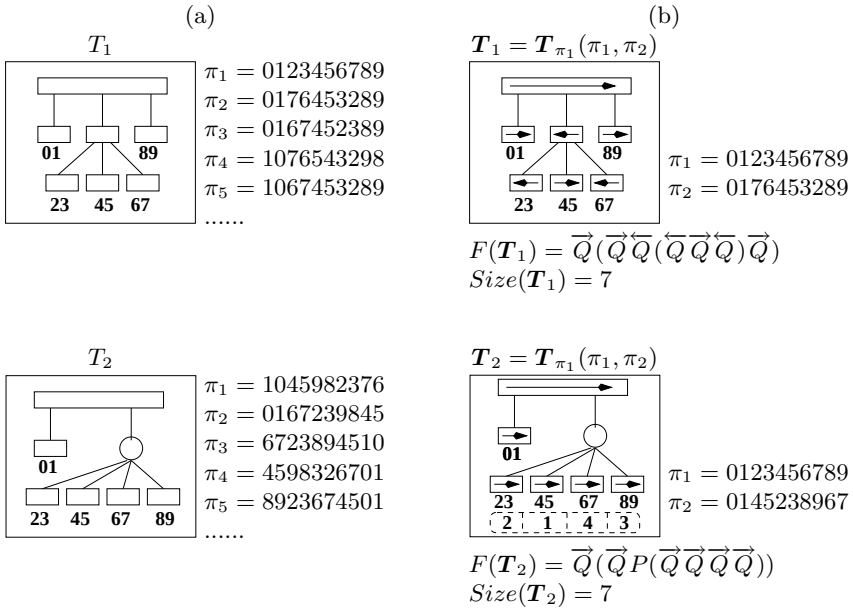


Fig. 2. Two examples shown in each row. (a) A PQ tree and a subset of the collection of permutations represented by the tree. (b) A *nailed* and *oriented* PQ tree and the only two permutations it represents.

Figure 2 shows two examples of oriented PQ trees and how they succinctly describe a pair of permutations. A *frontier*, $F(\boldsymbol{T})$, of a nailed and oriented tree $\boldsymbol{T}$ is simply the in-order notation of the PQ tree excluding the labeled leafnodes, with the orientation of the Q nodes denoted by a left or right arrow. Further, two nailed and oriented trees $\boldsymbol{T}$ and $\boldsymbol{T}'$ are equivalent, denoted as $\boldsymbol{T} \equiv \boldsymbol{T}'$, if and only if $F(\boldsymbol{T}) = F(\boldsymbol{T}')$. Notice that the leaf nodes (which are labeled 0-9 in Figure 2) are ignored while checking the equivalency of oriented PQ trees. The size of $\boldsymbol{T}$, denoted as, $Size(\boldsymbol{T})$ is the number of all the internal nodes (including the root). See Figure 2 for examples of frontier and size of some $\boldsymbol{T}$'s.

Next we explore the time to construct these PQ trees and the following is a direct consequence of the algorithm described in [28].

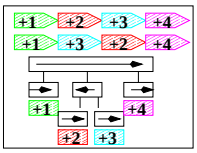
Theorem 1 ([28]). *Given two permutations π_1, π_2 of length n each, $\boldsymbol{T}_{\pi_1}(\{\pi_1, \pi_2\})$ can be constructed in $\mathcal{O}(n)$ time.*

Recall that $D^r(\pi_1, \pi_2)$ denotes the smallest inversion distance between π_1 and π_2. Let $D^t(\pi_1, \pi_2)$ denote the shortest translocation distance between the two and let $D(\pi_1, \pi_2)$ denote the shortest number of operations, inversion or translocation, that takes π_1 to π_2. The following theorem is central to the proposed algorithm.

Theorem 2. *Let π_1 and π_2 of size n each, be such that $D(\pi_1, \pi_2) = c$, for some constant c. Then there exists only $O(1)$ non-equivalent trees $\boldsymbol{T}(\pi_1, \pi_2)$, each of size $O(1)$.*

Outline of the proof: Let k be the maximum number of distinguishable segments that the input permutations can be split into by the rearrangement operation. Figure 3 shows the $c = 1$ case: for translocation operation, $k = 4$ and for inversion operation, $k = 3$ [2]. It is easy to see that k is independent of n and only

[2]



Left: a possibility with the maximum number of segments (four) using one translocation.

Right: a possibility with the maximum number of segments (three) using one inversion.

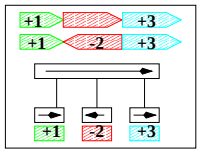
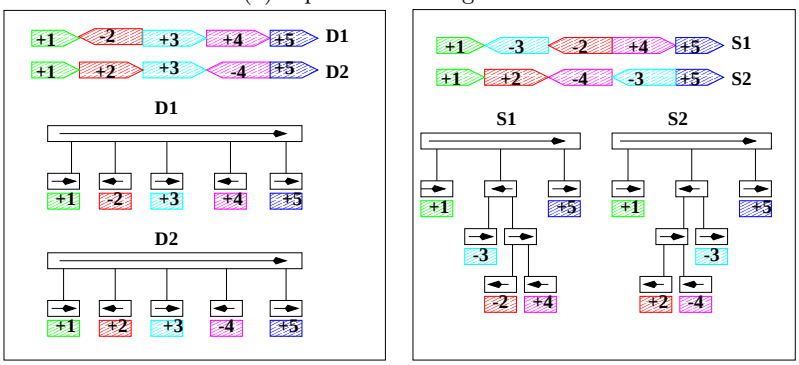
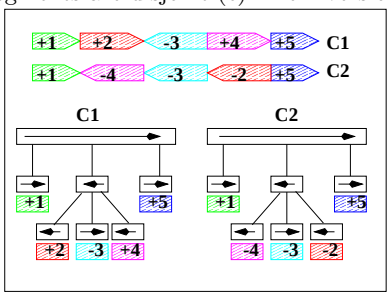


Fig. 3. Permutations at distance 1 from each other. Treating the segments as symbols, the left gives $\pi_1 = 1234$ and $\pi_2 = 1324$, with $\boldsymbol{T}(\pi_1, \pi_2)$ as shown and the right gives $\pi_1 = 123$ and $\pi_2 = 1(-2)3$, with $\boldsymbol{T}(\pi_1, \pi_2)$ as shown. The signed segment does not mean that the individual markers are signed. The algorithm to compute $\boldsymbol{T}_{\pi_1}(\{\pi_1, \pi_2\})$ detects the inverted order of the unsigned markers and annotates the segments accordingly.

(a) Input order of segments



(b) The inversion segments are disjoint (c) The inversion segments straddle



(d) One inversion segment is contained in the other

Fig. 4. (a) The given order of segments (the parent that will be computed from two separate inverted segments in each case that follow). The only three possible configurations of two inversion operations are shown in (b) to (d). (b) The two segments marked 2 and 4 on the input are *disjoint*. Labeling the two resulting permutations as $D1$ and $D2$, the first is $\boldsymbol{T}_{D1}(D1, D2)$ and the second is $\boldsymbol{T}_{D2}(D1, D2)$. (c) The two segments, marked 2-3 and 3-4 on the input, *straddle*. Labeling the two resulting permutations as $S1$ and $S2$, the first is $\boldsymbol{T}_{S1}(S1, S2)$ and the second is $\boldsymbol{T}_{S2}(S1, S2)$. (d) The two segments, marked 3 and 2-4 on the input are *nested*. Labeling the two resulting permutations as $C1$ and $C2$, the first is $\boldsymbol{T}_{C1}(C1, C2)$ and the second is $\boldsymbol{T}_{C2}(C1, C2)$.

dependent on c, thus $k = O(1)$. The number of distinct configurations depends only on k. Hence there can only be $O(1)$ distinct configurations.

For each distinct case, consider $\boldsymbol{T}(\pi_1, \pi_2)$. The leaves of $\boldsymbol{T}$ are partitioned into k sets, and thus the number of internal nodes is $\leq k$. Thus $Size(\boldsymbol{T}) = O(k) = O(1)$. □

2.1 Reconstructing ancestor permutations

Definition 4. ***(parent** $P_c(\Pi)$**)*** $(\pi' \notin \Pi) \in P_c(\Pi)$, *is a permutation such that for each* $\pi \in \Pi$, $D(\pi, \pi') \leq c$ *for some integer* $c \geq 0$.

Consider the task of computing $P_c(\Pi)$ where $\Pi = \{\pi_1, \pi_2\}$. If $D(\pi_1, \pi_2) = c$, then for each $\pi_p^i \in P_c(\Pi)^i$ is such that $D(\pi_1, \pi_p^i) = c_i$ and $D(\pi_2, \pi_p^i) = c - c_i$, for some $0 \leq c_i \leq c$. We first estimate $|P_c(\Pi)|$ in the following theorem.

Theorem 3. *Given $\pi_1 \neq \pi_2$, and a constant c, $|P_c(\pi_1, \pi_2)| = O(1)$.*

Outline of the proof: The above is a direct consequence of Theorem 2: the parents can be computed from distinct configurations which are $O(1)$ in number. Further each can give rise to only $O(1)$ parents, hence the result. □

We illustrate the use of a tree $\boldsymbol{T}(\pi_1, \pi_2)$ to compute a common parent through a simple example. For simplicity assume $c = 1$ and the only operation permitted is inversion. In Figure 4, we show the only possible three cases. The $\boldsymbol{T}$ shown are also called *masks* since they can be mechanically compared to the consensus nailed, oriented PQ trees of the given permutations. For clarity of exposition, each mask is shown in the two possible forms, when the resulting oriented PQ tree is nailed w.r.t π_1 and then w.r.t π_2. The algorithm to match the oriented PQ tree with a mask is outlined in Figure 5. In the algorithm the data structure for $\boldsymbol{T}$ is as follows: (1) $\boldsymbol{T}$.type is $\overleftarrow{Q}$, $\overrightarrow{Q}$ or P, (2) $\boldsymbol{T}.noc$ is the number of children of the node, (3) $\boldsymbol{T}.chld[i]$ is the pointer to the ith child of the node, and (4) $\boldsymbol{T}.Lvs$ is the leaves of the node, if the children of the node are leaves (else this is empty). This is best explained through an example.

This algorithm takes $O(1)$ time.

```
Match(T_c, T_m, ans)
 IF ((T_c.Lvs ≠ φ) AND (T_m.Lvs ≠ φ)) ans ← T_m.Lvs
 ELSE {
  ans ← φ, k_c ← T_c.noc, k_m ← T_m.noc
  IF (k_c ≤ k_m) {                                  //plausible match
   FOR l = 1 ... (k_m-k_c+1) {
    i_m ← 1, i_c ← l, lans ← φ
    WHILE (i_m < k_m) AND (i_c < k_c){
     IF (T_c.chld[i_c++].Type = T_m.chld[i_m++].Type)
      Match(T_c.chld[i_c-1], T_m chld[i_m-1], tans)
      lans ← lans + tans                            //concatenate the ans
    } //while
    IF (ans = φ) ans ← tans ELSE ans ← ans + "," + tans
   } //for
   ans ← "{" + ans + "}"
  } //if
 } //else
```

Fig. 5. The outline of the algorithm that matches a candidate $\boldsymbol{T}_c$ with the mask $\boldsymbol{T}_m$.

Consider Figure 9(a-c): The nailed, oriented PQ tree in (a) and (b) do not match any masks. However (c) matches mask D1 (or D2) of Figure 4. By the matching the first three segments, marked +1, +2 and +3 (0, 23451, 6) are placed in the same order and the fourth segment, marked -4 (789) is reversed giving the parent 0234516987. Thus the mask can be used to reconstruct a common parent. Figure 6 illustrates the use of masks on some simple examples that involve both inversions and translocations.

3 The Permutation Tree Construction Problem

Given a collection Π of members where each is defined by a sequence of markers, it is a natural question to reconstruct the genealogy ("evolutionary") tree.

Definition 5. *(Permutation Tree $\mathcal{T}$) Given Π a collection of m permutations on n integers, let $T(V, E)$ be a labeled tree where each node $v \in V$ is labeled by a permutation on n integers denoted as $\pi(v)$. Let $V' = \{(v \in V) \mid \pi(v) \in \Pi\}$. $\mathcal{T}(V, E)$ is a permutation tree on Π if the following conditions hold: (1) Each*

π_1	π_2	Mask $\boldsymbol{T}$	π_p
ab $\boxed{\text{hgfedc}}$ ijkl	abcde $\boxed{\text{kjihgf}}$ l	Fig 4(c)	abcdefghijkl
abcd $\boxed{\text{ef}}$ ghijkl	ab$\overline{ij}$cdfegh $\boxed{\uparrow}$ kl	Fig 7(a)	abcdfeghijkl
ab$\overline{ghij}$cdef $\boxed{\uparrow}$ klmn	abcd$\overline{ijkl}$efgh $\boxed{\uparrow}$ mn	Fig 7(b)	abcdefghijklmn
ab $\boxed{\text{fedc}}$ ghijkl	abcd$\overline{ij}$efgh $\boxed{\uparrow}$ kl	Fig 7(c)	abcdefghijkl
abcd$\overline{gh}$ef $\boxed{\uparrow}$ ijklmn	ab$\overline{kl}$cdefghij $\boxed{\uparrow}$ mn	Fig 7(d)	abcdefghijklmn
abc $\boxed{\text{fed}}$ ghijkl	a $\boxed{\text{kjihgfedcb}}$ l	Fig 4(d)	abcdefghijkl

Fig. 6. Reconstruction of common parent: A pair of permutations π_1 and π_2 and their common immediate parent π_p is shown here. An inversion is shown by a box and a translocation is shown as segment with a top bar being translocated to a destination shown by a boxed arrow. Although the operation is being shown here on each π_1 and π_2 for convenience, the same can be viewed as operations on the parent π_p that generates π_1 and π_2. A pointer to the PQ trees (masks) that are used to reconstruct the common parent is given in the last column.

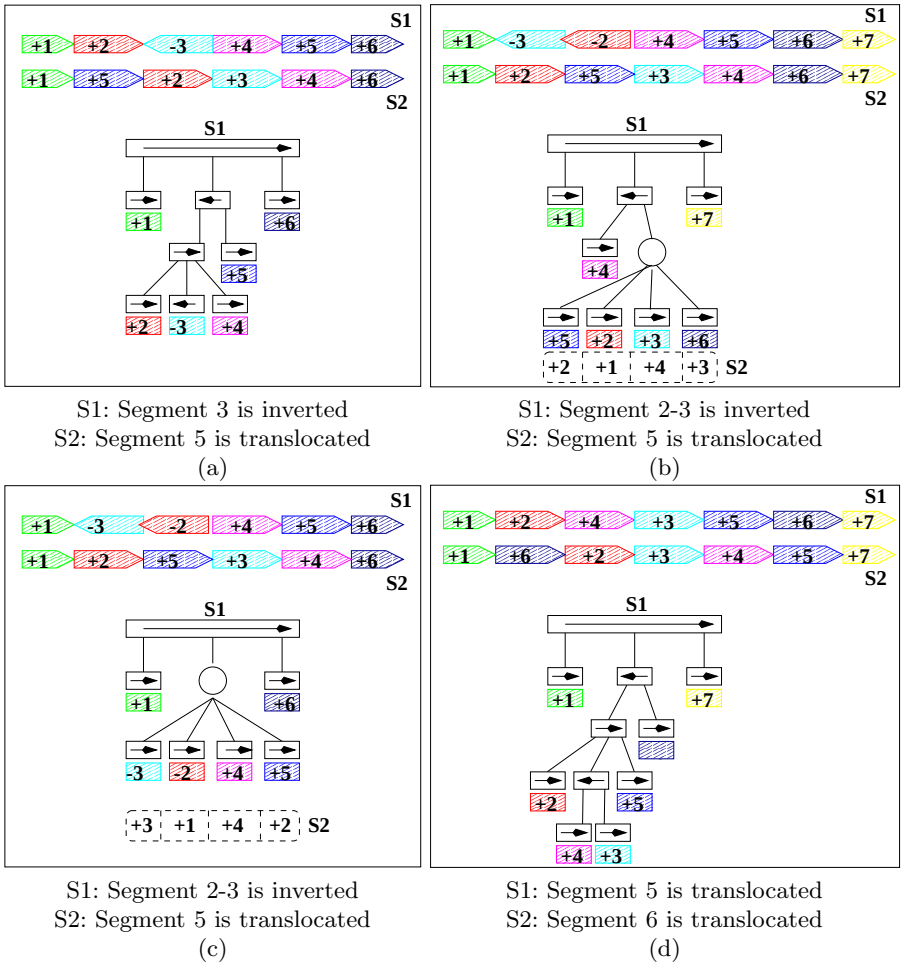


Fig. 7. In (a) and (c) the parent segment is +1+2+3+4+5+6 and in (c) and (d) parent segment is +1+2+3+4+5+6+7. In each of the cases, $\boldsymbol{T}_{S1}(\{S1, S2\})$ is displayed.

$v \in V'$ is labeled bijectively by the elements of Π, (2) for each leaf node $l \in V$, $\pi(l) \in \Pi$, (3) for each $(v_1v_2) \in E$, $\pi(v_1) \neq \pi(v_2)$ and (4) an internal node v that has degree < 3, must be such that $\pi(v) \in \Pi$.

Notice that the extant member can also be at an intermediate node of the tree unlike in most evolutionary tree construction problems [22] where the extant members can only appear as leaf nodes. Also this permutation

tree is unrooted. Further, let the length of the permutation tree $\mathcal{T}(V, E)$ be defined as

$$Len(\mathcal{T}) = \sum_{(v_1 v_2) \in E} D(\pi(v_1), \pi(v_2))$$

Problem 1 *(Permutation Tree Construction -PTC- Problem) Given Π, a collection of m permutations of length n each, the PTC problem is to construct the permutation tree $\mathcal{T}(V, E)$ of minimum length.*

Theorem 4. *The PTC problem is NP-complete.*

Outline of the proof: This is easily shown by contradiction. Assume there is a polynomial time solution to the PTC problem. Consider the special case where $\Pi = \{\pi_1, \pi_2\}$ and the only operations are inversions. Then $Len(\mathcal{T}) = D^r(\pi_1, \pi_2)$, but computing $D^r(\pi_1, \pi_2)$ is known to be NP-complete [6]. Hence the result. □

Perhaps a natural restriction is to use signed permutations, instead of unsigned ones, since there exist polynomial time algorithms to compute $D^r(\pi_1, \pi_2)$ [23, 27]. However, it is unclear how the common ancestors can be computed on the genealogy tree and heuristic methods have been proposed in literature for this problem [33, 4]. Since our problem is motivated by STRP's on human chromosomes, it is reasonable to assume that for each $(v_1 v_2) \in E$ of $\mathcal{T}(V, E)$, $D(\pi(v_1), \pi(v_2)) \leq c$, for some tiny constant c. We call this the cPTC problem and show that finding the exact solution to the problem is computationally tractable.

Problem 2 *(The cPTC Problem) The PTC problem with the added constraint that for each $(v_1 v_2) \in E$, $D(\pi(v_1), \pi(v_2)) \leq c$, for some small constant c.*

Notice that the problem definition allows for multifurcating trees. If the permutation tree exists, then Π is said to be *compatible*. If no such tree exists, the harder problem is to modify Π to make it compatible.

Problem 3 *(The near-cPTC Problem) Given $\Pi = \{\pi_1, \pi_2, \ldots \pi_m\}$, the near-cPTC problem is to minimize $\sum_{i=1}^{m} (D(\pi_i, (\pi_i' \in \Pi'))$ such that Π' is compatible.*

We claim that the PTC is a reasonable definition of the problem by proving that a randomly generated Π is seldom compatible. We first state two lemmas that lead to the theorem.

Lemma 1. *Given a permutation π of length n, $|\{(\pi' \neq \pi) \mid D(\pi', \pi) = 1\}| < n^q$, for some constant q.*

It is easy to see that $q = 2$ for the inversion operation and $q = 3$ for the translocation operation.

Lemma 2. *Let Π_i, $1 \leq i \leq k$, be k sets of independently chosen random permutations. If $\pi_i \in P_1(\Pi_i)$ then each π_i is independent of π_j, $i \neq j$.*

Theorem 5. *Given a random collection Π of m permutations of size n each, the expected number of permutation trees on Π is $o(1)$.*

Outline of the proof: Using Lemma 1 we compute the following probability where q is some constant:

$$\mathrm{P}(\pi \text{ is the immediate parent of some } \pi_1 \text{ and } \pi_2) = \frac{n^q}{n!^2}$$

Given a tree $\mathcal{T}$ with m leaf nodes, the probability $\mathrm{P}^{\Pi}(\mathcal{T})$ of it being a permutation tree on Π, $\mathrm{P}^{\Pi}(\mathcal{T}) = \left(\frac{n^q}{n!^2}\right)^m$ by Lemma 2. Further, N, the number of possible trees configurations is[3]:

$$N \leq \frac{(2m-4)!}{2^{m-2}(m-2)!}$$

By linearity of expectation, the expected number of trees on Π is bounded by $N\mathrm{P}^{\Pi}(\mathcal{T})$, which is $o(1)$ since $m \leq n!$. □

[3] This the the number of strictly bifurcating trees on m leaves and is a lower bound on all possible trees.

3.1 Algorithm

Here we present a very simple branch and bound algorithm to solve the cPTC problem (see Problem 2) using PQ trees.
Input: Π, a set of m permutations of size n each.
Output: A minimum length tree $\mathcal{T}(V, E)$ and a mapping $P : (v \in V) \rightarrow \Pi^*$, sending $(v \in V) \mapsto (\pi \in \Pi^*)$, where $\Pi \subset \Pi^*$.

In the description of the algorithm in Figure 8, let $\mathbf{A}$ denote an array of encoding of trees, indexed by permutation π. Thus $\mathbf{A}[\pi']$ stores the subtree rooted at v with $\pi(v) = \pi'$ encoded in the postfix notation as $s(\pi')$. Also assume that the + operator works as follows: $+(\Pi) = \pi'$ where $\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$ and $s(\pi') = \mathbf{A}(\pi_1)\mathbf{A}(\pi_2)\ldots\mathbf{A}(\pi_k)k\pi'$, the subtree rooted at π' in the postfix notation. $Lpi(\pi)$ is the collection of nodes labeled by $\pi' \in \Pi$ reachable from node labeled with π. $Chld(\pi)$ is the collection of immediate children of π. For the sake of clarity of exposition, the outline of the algorithm in Figure 8 excludes some implementation details.

```
Initialize
  Π_w ← Π
  FOR EACH π ∈ Π  Lpi(π) ← {π}, Chld(π) ← φ


WHILE (Π_w ≠ φ) {
    Π_new ← φ
    FOR EACH π_1, π_2 ∈ Π_w
        IF (Lpi(π_1) ∩ Lpi(π_2) = φ)                    //prune acyclic structures
            [S = P_c(π_1, π_2)]                          //compute common parents
            FOR EACH π ∈ S
                IF (π ∉ Π_w) Π_new ← Π_new ∪ {π}, Chld(π) ← φ
                IF (π ≠ π_1) Chld(π) ← Chld(π) ∪ {π_1}
                IF (π ≠ π_2) Chld(π) ← Chld(π) ∪ {π_2}
    FOR EACH π ∈ Π_new
        A[π] ← s(+(Chld(π)))                         //update subtrees of new nodes
    IF (Π_new = φ) Π_w ← φ                            //terminate if no new π's
        ELSE Π_w ← Π_w ∪ Π_new                        //add the new ones
}
```

Fig. 8. The outline of the algorithm to compute the permutation tree $\mathcal{T}$.

The algorithm works by computing common parents of the permutations. It continues the process till no more common parents can be computed. Since the common parents are not unique and there may be multiple trees, the algorithm keeps track of all possible trees in $\mathbf{A}[]$. Thus at the end of the loop for each π with $Lpi(\pi) = \Pi$, $s(\pi)$ denotes a plausible evolutionary tree.
Time Complexity. The masks are used in the boxed line of the algorithm in Figure 8 and the computation for each parent takes only $O(1)$ time, by Theorems 2 and 3. The minimal consensus PQ tree takes $O(n)$ time, by Theorem 1. Any permutation tree $\mathcal{T}$ on Π can have at most $O(m)$ nodes where $|\Pi| = m$. Also the number of iterations is $O(m)$. Thus the algorithm takes $O(m^3 n)$ time where , n is the length of each permutation $\pi \in \Pi$, provided no spurious trees are generated.
Concrete Example. An example with 8 permutations on 10 markers along with the permutation tree $\mathcal{T}$ is shown in Figure 10. For this example, the task of finding common parents using PQ trees is illustrated in Figure 9 in a few cases. The overall trace of the algorithm is shown in Figure 11.

4 Mutations

We will assume that the permutation on the markers will also include the specific allelic form it represents, i.e., say the copy number in case of micro satellites and the nucleic acid base in case of SNP's (Single

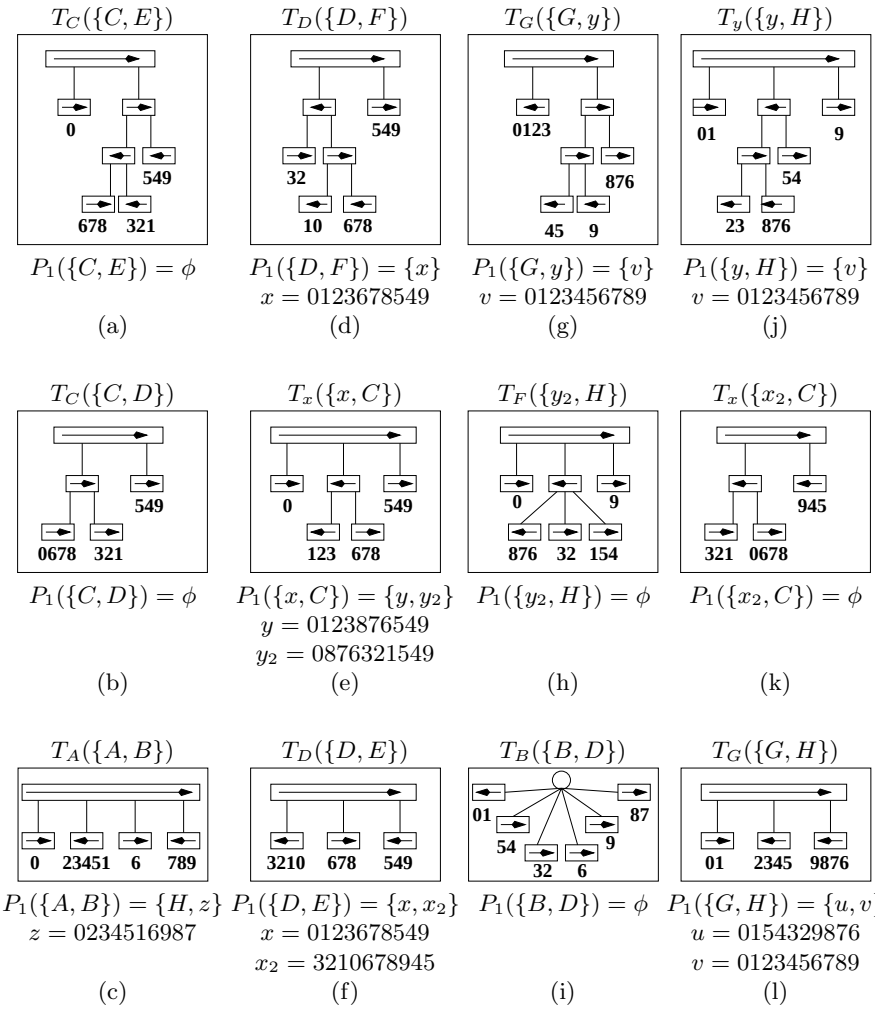


Fig. 9. Some computations (randomly chosen 12 parent computations) on the input permutations of the example of Figure 10. Permutations *A*-*G* are given and permutations u, v, x, x_2, y, y_2, z are computed in the intermediate steps. Consider (a): The boxed PQ tree is the minimal consensus PQ tree of $\Pi = \{C, E\}$ nailed with respect to C and the parent at distance 1 is given as $P_1(\Pi)$. (b)-(l) are to be similarly interpreted.

Nucleotide Polymorphism). Let $D^a(\pi_1, \pi_2)$ denote the number of markers that differ in their allelic form. For example, if $\pi_1 = 1^a2^a3^b4^c5^a$, $\pi_2 = 1^a3^a2^a5^c4^c$, where the superscript denotes an encoding of the allelic form, then $D^a(\pi_1, \pi_2) = 2$ since markers 3 and 5 vary in their allelic forms.

The proposed approach can be extended to include mutations and we are currently exploring this direction. In fact, in practice, the problem may be simplified by the use of mutations since, this will help time-order the events. We propose a two-step approach to the problem: first reconstruct the genealogy tree without using mutations. In the second step the tree can be resolved using the mutation information. The reason for taking this two-step approach is that the assumption that $D^a()$ is small is no longer valid under mutations. We plan to experiment on synthetic data to validate this approach.

5 Conclusion

Here we present a systematic way of studying large scale genome rearrangements to construct a genealogy tree within a species. The problem is motivated by the recent discoveries of inversion and translocations within the human population. The approach is based on our earlier work on computing minimal consensus PQ trees of permutations along with the observation that when the edit distances are small, only $O(1)$

number of PQ trees of size $O(1)$ each need to be considered. This gives an efficient algorithm to compute the underlying genealogy tree and the reconstruction of all the common ancestors.

Acknowledgments. I would like to thank Oren and Enam for their efforts with the implementation of the consensus PQ tree construction.

References

1. A. Bergeron. A very elementary presentation of the hannenhalli-pevzner theory. In *Proc. of the Twelfth Symp. on Comp. Pattern Matching*, volume 2089 of *Lecture Notes in Computer Science*, pages 106–117. Springer-Verlag, 2001.
2. A. Bergeron and J. Stoye. On the similarity of sets of permutations and its applications to genome comparison. In *Proc. of COCOON*, volume 2089 of *Lecture Notes in Computer Science*, pages 68–79. Springer-Verlag, 2003.
3. S. Bloom. Using genetics to unearth our path on earth. *J. of. Clinical Investigation*, 115:1395, 2005.
4. G. Bourque and P. A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. In *Genome Research*, pages 26–36. Cold Spring Harbor Laboratory Press, 2002.
5. R. L. Cann, M. Stoneking, and A. C. Wilson. Mitochondrial DNA and human evolution. *Nature*, 356:389–390, 1992.
6. A. Caprara. Formulations and complexity of multiple sorting by reversals. In *Proceedings of the Annual Conference on Computational Molecular Biology (RECOMB99)*, pages 84–93. ACM Press, 1999.
7. The Intrnational HapMap Consortium. The International HapMap Project. *Nature*, 426:789–796, 2003.
8. T. Dobzhansky and A. Sturtevant. Inversions in the chrnomosomes of drosophila pseudoobscura. *Genetics*, 23:28–64, 1938.
9. Revital Eres, Gad Landau, and Laxmi Parida. A combinatorial approach to automatic discovery of cluster-patterns. In *Proc. of the Third Wrkshp. on Algorithms in Bioinformatics*, volume 2812 of *Lecture Notes in Bioinformatics*, pages 139–150. Springer-Verlag, 2003.
10. A. Kakizuka et al. Chromosomal translocation t(15;17) in human acute promyelocytic leukemia fuses rar alpha with a novel putative transcription factor, PML. *Cell*, 66(4):663–674, 1991.
11. A. Kakizuka et al. Identification of novel genes, SYT and SSX, involved in the t(X;18)(p11.2;q11.2) translocation found in human synovial sarcoma. *Nature Genetics*, 7:502–508, 1994.
12. C. A. Tilford et. al. A physical map of the human Y chromosome. *Nature*, 409:943–945, 2001.
13. D. Lakich et. al. Inversions disrupting the factor VIII gene are a common casue of severe haemophilia. *Nature Genetics*, 5:236–241, 1993.
14. H. Kehrer-Sawatzki et. al. Molecular characterizations of the pericentric inversion that causes difference between chimpanzee chromosome 19 and human chromosome 17. *Am J. of Hum. Genetics*, 71:375–388, 2002.
15. H. Stefansson et. al. A common inversion under selection in europeans. *Nat. genetics*, 37(2):129–137, 2005.
16. M. E. Cosner et. al. An empirical comparison of phylogenetic methods on chloroplast gene order data in campanulaceae. *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*, 2000.
17. M. J. Pettenati et. al. Paracentric inversions in humans: a review of 446 paracentric inversions with presentations of 120 new cases. *Am J. of Med. Genetics*, 55:171–187, 1995.
18. M. L. Bondeson et. al. Inversion of the IDS gene resulting from recombination with IDS-related sequences is a common cause of the Hunter syndrome. *Molecular Genetics*, 4:615–621, 1995.
19. M. T. Pletcher et. al. Use of comparative physical and sequence mapping to annotate mouse chromosome 16 and human chromosome 21. *Genomics*, 74:45–54, 2001.
20. S. Giglio et. al. Olfactory receptor-gene clusters, genomic inversion polymorphisms and common chromosome rearrangements. *Am J. of Hum. Genetics*, 68:874–883, 2001.
21. J. Felsenstein. *Inferring phylogenies.* Sinauer Associates, 2004.
22. D. Gusfield. *Algorithms on strings, trees and sequencess: computer science and computational biology.* Cambridge University Press, New York, 1997.
23. S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. In *J. of ACM*, volume 46, pages 1–27. ACM Press, 1999.
24. J. J. Hudson. *Gene genalogies and the coalescent process.* Oxford Surveys in Evolutionary Biology. Oxford University Press, Oxford, 1990.
25. M. A. Huynen, B. Snel, and P. Bork. Inversions and the dynamics of eukaryotic gene order. *Trends in Genetics*, 17:304–306, 2001.
26. Booth K. and Leukar G. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

27. H. Kaplan, R. Shamir, and R. E. Tarjan. A faster and simpler algorithm for sorting signed permutations by reversals. In *SIAM J. of Computing*, volume 29, pages 880–892, 1999.
28. Gad Landau, Laxmi Parida, and Oren Weimann. Using pq trees for comparative genomics. In *Proc. of the Symp. on Comp. Pattern Matching*, volume 3537 of *Lecture Notes in Computer Science*, pages 128–143. Springer-Verlag, 2005.
29. J. Palmer. Chloroplast and mitochondrial genome evolution in land plants. *Cell Organelles*, pages 99–133, 1988.
30. J. Palmer and I. Hebron. Plant mitochondrial DNA evolves rapidly in structure but slowly in sequence. *J. Mol. Evol.*, 27:87–97, 1988.
31. D. Sankoff. Edit distance for genome comparison based on non-local operations. In *Proc. of the Third Symp. on Comp. Pattern Matching*, pages 121–135. Springer-Verlag, 1992.
32. D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *Journal of Computational Biology*, 5(3):555–570, 1998.
33. D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proc. Nat. Acad. Sci.*, 89:6575–6579, 1992.
34. K. Small, J. Iber, and S. T.Warren. Emerin deletion reveals a common X-chromosome inversion mediated by inverted repeats. *Nature Genetics*, 16:96–99, 1997.
35. M.S. Waterman. *An Introduction to Computational Biology: Maps, Sequences and Genomes*. Chapman Hall, 1995.
36. www.nationalgeographic.com/genographic. 2005.

	Π
A	0234516789
B	0154326987
C	0678321549
D	3210678549
E	0123678945
F	0187632549
G	0123459876
H	0154326789

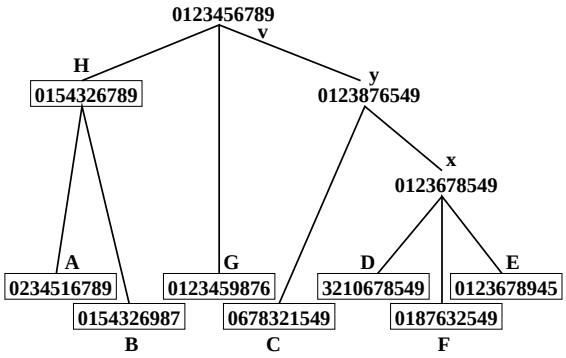


Fig. 10. Π shown on the left. Each member is a permutation on ten markers numbered 0 through 9. On the right, the genealogy tree on Π is shown with the members of Π enclosed in boxes.

Π	iteration 1	iteration 2	iteration 3
A=0234516789	$H \in P_1(A, B)$		
B=0154326987	$(z \in P_1(A, B))$=0234516987		
H=0154326789			$(v \in P_1(H, y, G))$
C=0678321549		$(y \in P_1(C, x))$	= 0123456789
D=3210678549	$(x \in P_1(D, E, F))$=0123678549	= 0123876549	
E=0123678945	$(x_2 \in P_1(D, E))$=3210678945	$(y_2 \in P_1(C, x))$	$(u \in P_1(H, G))$
F=0187632549	$(x_3 \in P_1(E, F))$=0187632945	= 0876321549	= 0154329876
G=0123459876			

(1)

π	$s(\pi)$	$Lpi(s(\pi))$
x	$DEF3x$	D, E, F
x_2	$DE2x_2$	D, E
x_3	$EF2x_3$	E, F
z	$AB2z$	A, B
y	$CDEF3x2y$	C, D, E, F
y_2	$CDEF3x2y_2$	C, D, E, F
u	$AB2HG2u$	A, B, G, H
v	$AB2HCDEF3x2yG2v$	A, B, C, D, E, F, G, H

(2)

Fig. 11. (1) The trace of the algorithm through three iterations on Π shown in the first column. Only non-empty parent computations are shown here. (2) The table shows different values of the computed intermediate permutations, π and $s(\pi)$ and $Lpi(\pi)$. Clearly, v is the valid evolutionary tree since $Lpi(v) = \Pi$.