# Topological Rearrangements and Local Search Method for Tandem Duplication Trees

Denis Bertrand, Olivier Gascuel

# Topological Rearrangements and Local Search Method for Tandem Duplication Trees

Denis Bertrand and Olivier Gascuel

**Abstract**—The problem of reconstructing the duplication history of a set of tandemly repeated sequences was first introduced by Fitch [4]. Many recent studies deal with this problem, showing the validity of the unequal recombination model proposed by Fitch, describing numerous inference algorithms, and exploring the combinatorial properties of these new mathematical objects, which are duplication trees. In this paper, we deal with the topological rearrangement of these trees. Classical rearrangements used in phylogeny (NNI, SPR, TBR, ...) cannot be applied directly on duplication trees. We show that restricting the neighborhood defined by the SPR (Subtree Pruning and Regrafting) rearrangement to valid duplication trees, allows exploring the whole duplication tree space. We use these restricted rearrangements in a local search method which improves an initial tree via successive rearrangements. This method is applied to the optimization of parsimony and minimum evolution criteria. We show through simulations that this method improves all existing programs for both reconstructing the topology of the true tree and recovering its duplication events. We apply this approach to tandemly repeated human Zinc finger genes and observe that a much better duplication tree is obtained by our method than using any other program.

**Index Terms**—Tandem duplication trees, phylogeny, topological rearrangements, local search, parsimony, minimum evolution, Zinc finger genes.

## 1 Introduction

Repeated sequences constitute an important fraction of most genomes, from the well-studied Escherichia coli bacterial genome [1] to the Human genome [2]. For example, it is estimated that more than 50 percent of the Human genome consists of repeated sequences [2], [3]. There exist three major types of repeated sequences: transposon-derived repeats, micro or minisatellites, and large duplicated sequences, the last often containing one or several RNA or protein-coding genes. Micro or minisatellites arise through a mechanism called slipped-strand mispairing, and are always arranged in tandem: copies of a same basic unit are linearly ordered on the chromosome. Large duplicated sequences are also often found in tandem and, when this is the case, unequal recombination is widely assumed to be responsible for their formation.

Both the linear order among tandemly repeated sequences, and the knowledge of the biological mechanisms responsible for their generation, suggest a simple model of evolution by duplication. This model, first described by Fitch in 1977 [4], introduces tandem duplication trees as phylogenies constrained by the unequal recombination mechanism. Although being a completely different biological mechanism, slipped-strand mispairing leads to the same duplication model [5]. A formal recursive definition of this model is provided in Section 2, but its main features can be grasped from the examples of Fig. 1. Fig. 1a shows the duplication history of the 13 Antennapedia-class homeobox genes from the cognate group [6]. In this history, the ancestral locus has undergone a series of **simple duplication events** where one of the genes has been duplicated into two adjacent copies. Starting from the unique ancestral gene, this series of events has produced the extant locus containing the 13 linearly ordered contemporary genes. It is easily seen [7] that trees only containing simple duplication events are equivalent to binary search trees with labeled leaves. They differ from standard phylogenies in that node children have left/right orientation. Fig. 1b shows another example corresponding to the nine variable genes of the human T cell receptor Gamma (TRGV) locus [8]. In this history, the most recent event involves a **double duplication** where two adjacent genes have been simultaneously duplicated to produce four adjacent copies. Duplication trees containing multiple duplication events differ from binary search trees, but are less general than phylogenies. The model proposed by Fitch [4] covers both simple and multiple duplication trees.

Fitch's paper [4] received relatively little attention at the time of its publication probably due to the lack of available sequence data. Rediscovered by Benson and Dong [9], Tang et al. [10], and Elemento et al. [8], tandemly repeated sequences and their suggested duplication model have recently received much interest, providing several new computational biology problems and challenges [11], [12]. The main challenge consists of creating algorithms incorporating the model constraints to reconstruct the

• *The authors are with Projet Méthodes et Algorithmes pour la Bioinformatique, LIRMM (UMR 5506, CNRS—Univ. Montpellier 2), 161 rue Ada, 34392 Montpellier Cedex 5—France. E-mail: gascuel@lirmm.fr.*

*Manuscript received 11 Oct. 2004; revised 17 Dec. 2004; accepted 20 Dec. 2004; published online 30 Mar. 2005.*
*For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-0170-1004.*

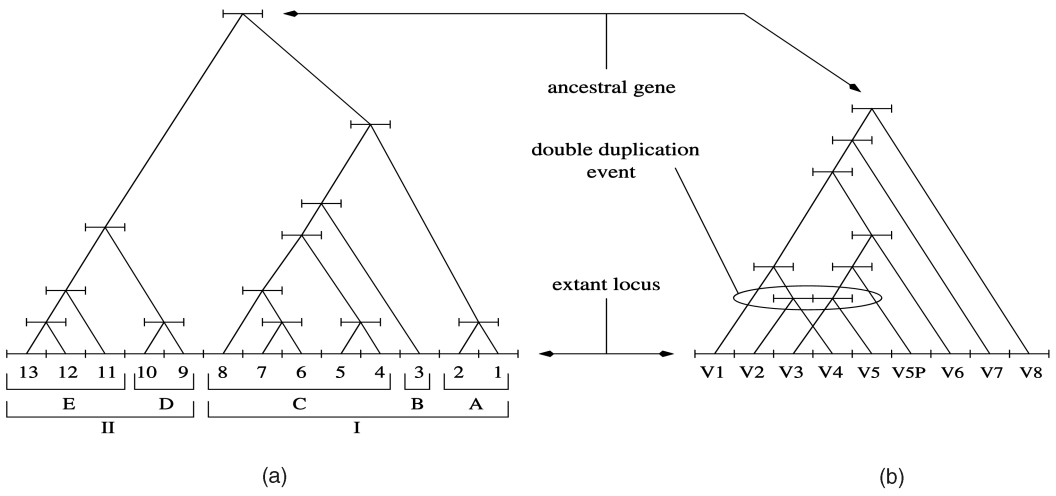



Fig. 1. (a) Rooted duplication tree describing the evolutionary history of the 13 Antennapedia-class homeobox genes from the cognate group [6]. (b) Rooted duplication tree describing the evolutionary history of the nine variable genes of the human T cell receptor Gamma (TRGV) locus [8]. In both examples, the contemporary genes are adjacent and linearly ordered along the extant locus.

duplication history of tandemly repeated sequences. Indeed, accurate reconstruction of duplication histories will be useful to elucidate various aspects of genome evolution. They will provide new insights into the mechanisms and determinants of gene and protein domain duplication, often recognized as major generators of novelty [13]. Several important gene families, such as immunity-related genes, are arranged in tandem; better understanding their evolution should provide new insights into their duplication dynamics and clues about their functional specialization. Studying the evolution of micro and minisatellites could resolve unanswered biological questions regarding human migrations or the evolution of bacterial diseases [14].

Given a set of aligned and ordered sequences (DNA or proteins), the aim is to find the duplication tree that best explains these sequences, according to usual criteria in phylogenetics, e.g., parsimony or minimum evolution. Few studies have focused on the computational hardness of this problem, and all of these studies only deal with the restricted version where simultaneous duplication of multiple adjacent segments is not allowed. In this context, Jaitly et al. [15] shows that finding the optimal single copy duplication tree with parsimony is NP-Hard and that this problem has a PTAS (Polynomial Time Approximation Scheme). Another closely related PTAS is given by Tang et al. [10] for the same problem. On the other hand, Elemento et al. [7] describes a polynomial distance-based algorithm that reconstructs optimal single copy tandem duplication trees with minimum evolution.

However, it is commonly believed, as in phylogeny, that most (especially multiple) duplication tree inference problems are NP-Hard. This explains the development of heuristic approaches. Benson and Dong [9] provides various parsimony-based heuristic reconstruction algorithms to infer duplication trees, especially from minisatellites. Elemento et al. [8] present an enumerative algorithm that computes the most parsimonious duplication tree; this algorithm (by its exhaustive approach) is limited to datasets of less than 15 repeats. Several distance-based methods have also been described. The WINDOW method [10] uses an agglomeration scheme similar to UPGMA [16] and NJ [17], but the cost function used to judge potential duplication is based on the assumption that the sequences follow a molecular clock mode of evolution. The DTSCORE method [18] uses the same scheme but corrects this limitation using a score criterion [19], like ADDTREE [20]. DTSCORE can be used with sequences that do not follow the molecular clock, which is, for example, essential when dealing with gene families containing pseudogenes that evolve much faster than functional genes. Finally, GREEDY SEARCH [21] corresponds to a different approach divided into two steps: First, a phylogeny is computed with a classical reconstruction method (NJ), then, with nearest neighbor interchange (NNI) rearrangements, a duplication tree close to this phylogeny is computed. This approach is noteworthy since it implements topological rearrangements which are highly useful in phylogenetics [22], but it works blindly and does not ensure that good duplication trees will be found (cf. Section 5.2).

Topological rearrangements have an essential function in phylogenetic inference, where they are used to improve an initial phylogeny by subtree movement or exchange. Rearrangements are very useful for all common criteria (parsimony, distance, maximum likelihood) and are integrated into all classical programs like PAUP* [23] or PHYLIP [24]. Furthermore, they are used to define various distances between phylogenies and are the foundation of much mathematical work [25]. Unfortunately, they cannot be directly used here, as shown by a simple example given

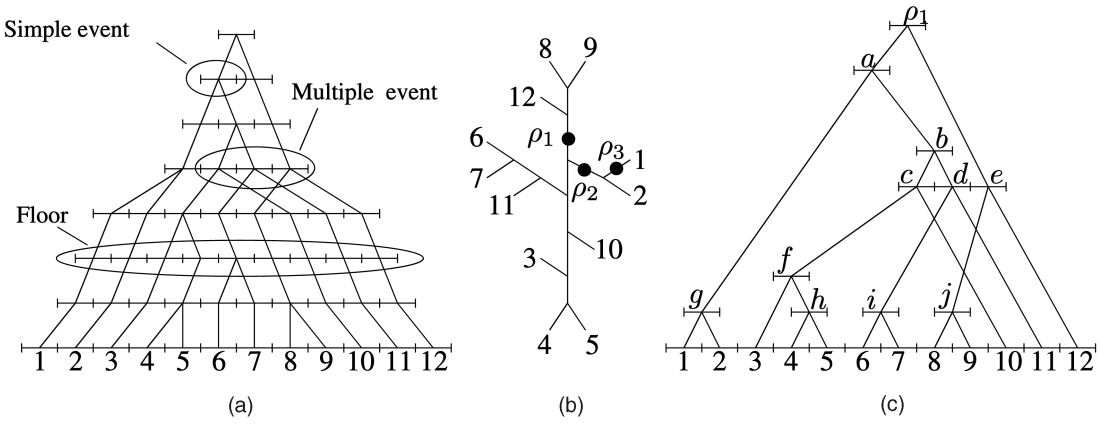



Fig. 2. (a) Duplication history; each segment represents a copy; extant segments are numbered. (b) Duplication tree (DT); the black points show the possible root locations. (c) Rooted duplication tree (RDT) corresponding to history (a) and root position $\rho_1$ on (b).

later. Indeed, when applied to a duplication tree, they do not guarantee that another valid duplication tree will be produced.

In this paper, we describe a set of topological rearrangements to stay inside the duplication tree space and explore the whole space from any of its elements. We then show the advantages of this approach for duplication tree inference from sequences. In Section 2, we describe the duplication model introduced by [4], [8], [10], as well as an algorithm to recognize duplication trees in linear time. Thanks to this algorithm, we restrict the neighborhoods defined by classical phylogeny rearrangements, namely, nearest neighbor interchange (NNI) and subtree pruning and regrafting (SPR), to valid duplication trees. We demonstrate (Section 3) that for NNI moves this restricted neighborhood does not allow the exploration of the whole duplication tree space. On the other hand, we demonstrate that the restricted neighborhood of SPR rearrangement allows the whole space to be explored. In this way, we define a local search method, applied here to parsimony and minimum evolution (Section 4). We compare this method to other existing approaches using simulated and real data sets (Section 5). We conclude by discussing the positive results obtained by our method, and indicate directions for further research (Section 6).

## 2 Model

### 2.1 Duplication History and Duplication Tree

The tandem duplication model used in this article was first introduced by Fitch [4] then studied independently by [8], [10]. It is based on unequal recombination which is assumed to be the sole evolution mechanism (except point mutations) acting on sequences. Although it is a completely different biological mechanism, slipped-strand mispairing leads to the same duplication model [5], [9].

Let $O = (1, 2, \ldots, n)$ be the ordered set of sequences representing the extant locus. Initially containing a single copy, the locus grew through a series of consecutive duplications. As shown in Fig. 2a, a **duplication history** may contain simple duplication events. When the duplicated fragment contains two, three, or $k$ repeats, we say that it involves a multiple duplication event. Under this duplication model, a duplication history is a rooted tree with $n$ labeled and ordered leaves, in which internal nodes of degree 3 correspond to duplication events. In a real duplication history (Fig. 2a), the time intervals between consecutive duplications are completely known, and the internal nodes are ordered from top to bottom according to the moment they occurred in the course of evolution. Any ordered segment set of the same height then represents an ancestral state of the locus. We call such a set a floor, and we say that two nodes $i, j$ are adjacent ($i \prec j$) if there is a floor where $i$ and $j$ are consecutive and $i$ is on the left of $j$.

However, in the absence of a molecular clock mode of evolution (a typical problem), it is impossible to recover the order between the duplication events of two different lineages from the sequences. In this case, we are only able to infer a **duplication tree (DT)** (Fig. 2b) or a **rooted duplication tree (RDT)** (Fig. 2c).

A duplication tree is an unrooted phylogeny with ordered leaves, whose topology is compatible with at least one duplication history. Also, internal nodes of duplication trees are partitioned into **events** (or "blocks" following [10]), each containing one or more (ordered) nodes. We distinguish "simple" duplication events that contain a unique internal node (e.g., $b$ and $f$ in Fig. 2c) and "multiple" duplication events which group a series of adjacent and simultaneous duplications (e.g., $c$, $d$, and $e$ in Fig. 2c). Let $E = (s_i, s_{i+1}, \ldots, s_k)$ denote an event containing internal nodes $s_i, s_{i+1}, \ldots, s_k$ in left to right order. We say that two consecutive nodes of the same event are **adjacent** ($s_j \prec s_{j+1}$) just like in histories, as any event belongs to a floor in all of

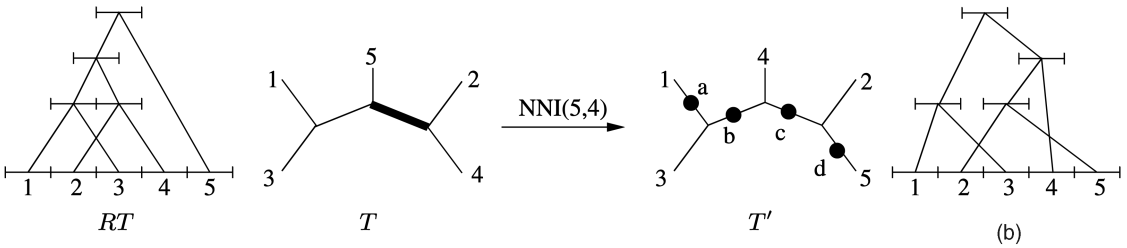



Fig. 3. The tree obtained by applying an NNI move to a DT is not always a valid DT: $T$ whose $RT$ is a rooted version; $T'$ is obtained by applying NNI(5,4) around the bold edge; none of the possible root positions of $T'$ (a, b, c, and d) leads to a valid RDT, cf. tree (b) which corresponds to root b in $T'$.

the histories that are compatible with the DT being considered. The same notation will also be used for leaves to express the segment order in the extant locus. When the tree is rooted, every internal node $s_j$ is unambiguously associated to one parent and two child nodes; moreover, one child of $s_j$ is "left" and the other one is "right," which is denoted as $l_j$ and $r_j$, respectively. In this case, for any duplication history that is compatible with this tree, child nodes of an event, $s_i, s_{i+1}, \ldots, s_k$ are organized as follows:

$$l_i \prec l_{i+1} \prec \ldots \prec l_k \prec r_i \prec r_{i+1} \prec \ldots \prec r_k.$$

In [8], [26], [27], it was shown that rooting a duplication tree is different than rooting a phylogeny: the root of a duplication tree necessarily lies on the tree path between the most distant repeats on the locus, i.e., 1 and $n$; moreover, the root is always located "above" all multiple duplications, e.g., Fig. 1b shows that there are only three valid root positions, the root cannot be a direct ancestor of 12.

### 2.2 Recursive Definition of Rooted and Unrooted Duplication Trees

A duplication tree is compatible with at least one duplication history. This suggests a recursive definition, which progressively reconstructs a possible history, given a phylogeny $T$ and a leaf ordering $O$. We define a **cherry** $(l, s, r)$ as a pair of leaves ($l$ and $r$) separated by a single node $s$ in $T$, and we call $C(T)$ the set of cherries of $T$. This recursive definition reverses evolution: It searches for a "visible duplication event," "agglomerates" this event, and checks whether the "reduced" tree is a duplication tree. In case of rooted trees, we have:

$(T, O)$ defines a duplication tree with root $\rho$ if and only if:
1. $(T, O)$ only contains $\rho$, or
2. there is in $C(T)$ a series of cherries $(l_i, s_i, r_i), (l_{i+1}, s_{i+1}, r_{i+1}), \ldots, (l_k, s_k, r_k)$ with $k \geq i$ and $l_i \prec l_{i+1} \prec \ldots \prec l_k \prec r_i \prec r_{i+1} \prec \ldots \prec r_k$ in $O$, such that $(T', O')$ defines a duplication tree with root $\rho$, where $T'$ is obtained from $T$ by removing $l_i, l_{i+1}, \ldots, l_k, r_i, r_{i+1}, \ldots, r_k$, and $O'$ is obtained by replacing $(l_i, l_{i+1}, \ldots, l_k, r_i, r_{i+1}, \ldots, r_k)$ by $(s_i, s_{i+1}, \ldots, s_k)$ in $O$.

The definition for unrooted trees is quite similar:

$(T, O)$ defines an unrooted duplication tree if and only if:
1. $(T, O)$ contains 1 segment, or
2. same as for rooted trees with $(T', O')$ now defining an unrooted duplication tree.

Those definitions provide a recursive algorithm, **RADT (Recognition Algorithm for Duplication Trees)**, to check whether any given phylogeny with ordered leaves is a duplication tree. In case of success, this algorithm can also be used to reconstruct duplication events: At each step, the series of internal nodes above denoted as $(s_i, s_{i+1}, \ldots, s_k)$ is a duplication event. When the tree is rooted, $l_j$ is the left child of $s_j$ and $r_j$ its right child, for every $j, i \leq j \leq k$. This algorithm can be implemented in $O(n)$ [26] where $n$ is the number of leaves. Another linear algorithm is proposed by Zhang et al. [21] using a top down approach instead of a bottom-up one, but applies only to rooted duplication trees.

## 3 Topological Rearrangements for Duplication Trees

This section shows how to explore the DT space using SPR rearrangements. First, we describe some NNI, SPR, and TBR rearrangement properties with standard phylogenies. But, these rearrangements cannot be directly used to explore the DT space. Indeed, when applied to a duplication tree, they do not guarantee that another valid duplication tree will be produced. So, we have decided to restrict the neighborhood defined by those rearrangements to duplication trees. If we only used NNI rearrangements, the neighborhood would be too restricted (as shown by a simple example) and would not allow the whole DT space to be explored. On the other hand, we can distinguish two types of SPR rearrangements which, when applied to a rooted duplication tree guarantee that another valid duplication tree will be produced. Thanks to these specific rearrangements, we demonstrate that restricting the neighborhood of SPR rearrangements allows the whole space of duplication trees to be explored.

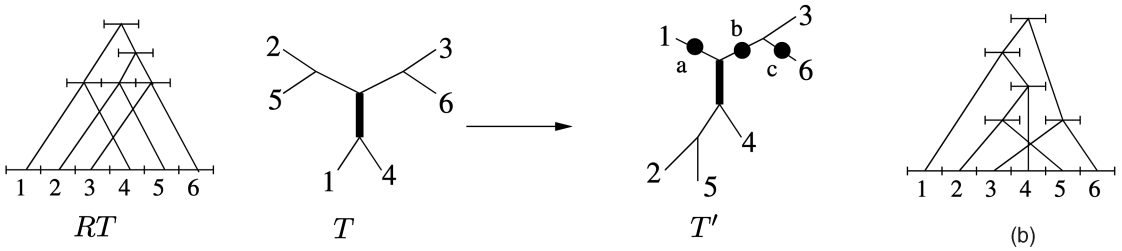



Fig. 4. The NNI neighborhood of a duplication tree does not always contain duplication trees: $T$ whose $RT$ is a rooted version; $T'$ is obtained by exchanging subtrees 1 and (2 5); none of the possible root positions of $T'$ (a, b, and c) leads to a valid duplication tree, cf. tree (b) which corresponds to root b in $T'$; and the same holds for every neighbor of $T$ being obtained by NNI.

### 3.1 Topological Rearrangements for Phylogeny

There are many ways of carrying out topological rearrangements on phylogeny [22]. We only describe **NNI** (Nearest Neighbor Interchange), **SPR** (Subtree Pruning Regrafting), and **TBR** (Tree Bisection and Reconnection) rearrangements.

The NNI move is a simple rearrangement which exchanges two subtrees adjacent to the same internal edge (Figs. 3 and 4). There are two possible NNIs for each internal edge, so $2(n-3)$ neighboring trees for one tree with $n$ leaves. This rearrangement allows the whole space of phylogeny to be explored; i.e., there is a succession of NNI moves making it possible to transform any phylogeny $P_1$ into any phylogeny $P_2$ [28].

The SPR move consists of pruning a subtree and regrafting it, by its root, to an edge of the resulting tree (Figs. 6 and 7). We note that the neighborhood of a tree defined by the NNI rearrangements is included in the neighborhood defined by SPRs. The latter rearrangement defines a neighborhood of size $2(n-3)(2n-7)$ [25].

Finally, TBR generalizes SPR by allowing the pruned subtree to be reconnected by any of its edges to the resulting tree. These three rearrangements (NNI, SPR, and TBR) are reversible, that is, if $T'$ is obtained from $T$ by a particular rearrangement, then $T$ can be obtained from $T'$ using the same type of rearrangement.

### 3.2 NNI Rearrangements Do Not Stay in DT Space

The classical phylogenetic rearrangements (NNI, SPR, TBR,...) do not always stay in DT space. So, if we apply an NNI to a DT (e.g., Fig. 3), the resulting tree is not always a valid DT. This property is also true for SPR and TBR rearrangements since NNI rearrangements are included in these two rearrangement classes.

### 3.3 Restricted NNI Does Not Allow the Whole DT Space to Be Explored

To restrict the neighborhood defined by NNI rearrangements to duplication trees, each element of the neighborhood is filtered thanks to the recognition algorithm (RADT). But, this restricted neighborhood does not allow the whole DT space to be explored. Fig. 4 gives an example of a duplication tree, $T$, the neighborhood of which does not contain any DT. So, its restricted neighborhood is empty, and there is no succession of restricted NNIs allowing $T$ to be transformed into any other DT.

### 3.4 Restricted SPR Allows the Whole DT Space to Be Explored

As before, we restrict (using RADT) the neighborhood defined by SPR rearrangements to duplication trees. We name **restricted SPR**, SPR moves that, starting from a duplication tree, lead to another duplication tree.

**Main Theorem.** *Let $T_1$ and $T_2$ be any given duplication trees; $T_1$ can be transformed into $T_2$ via a succession of restricted SPRs.*

**Proof.** To demonstrate the Main Theorem, we define two types of special SPR that ensure staying within the space of rooted duplication trees (RDT). Given these two types of SPRs, we demonstrate that it is possible to transform any rooted duplication tree into a **caterpillar**, i.e., a rooted tree in which all internal nodes belong to the tree path between the leaf 1 and the tree root $\rho$ (cf. Fig. 5).

This result demonstrates the theorem. Indeed, let $T_1$ and $T_2$ be two RDTs. We can transform $T_1$ and $T_2$ into a caterpillar by a succession of restricted SPRs. So, it is possible to transform $T_1$ into $T_2$ by a succession of restricted SPRs, with (possibly) a caterpillar as intermediate tree. This property holds since the reciprocal movement of an SPR is an SPR. As the two SPR types proposed ensure that we stay within the RDTs space, we have the desired result for rooted duplication trees. And, this result extends to unrooted duplications trees since two DTs can be arbitrarily rooted, transformed from one to the other using restricted SPRs, then unrooted. □

The first special SPR allows multiple duplication events to be destroyed. Let $E = (s_i, s_{i+1}, \ldots, s_k)$ be a duplication event, $r_i$ and $l_k$ respectively right child of $s_i$

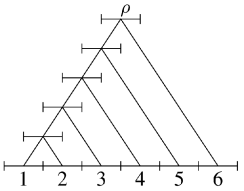



Fig. 5. A six-leaf caterpillar.

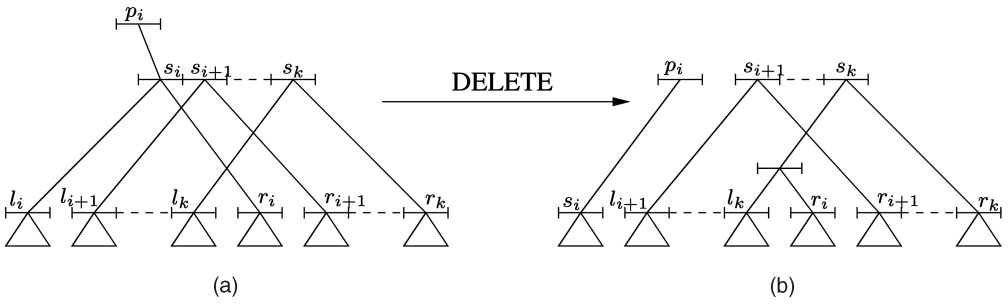



Fig. 6. DELETE rearrangement.

and left child of $s_k$, and let $p_i$ be the father of $s_i$. The **DELETE** rearrangement consists of pruning the subtree of root $r_i$ and grafting this subtree on the edge $(s_k, l_k)$, while $l_i$ is renamed $s_i$ and the edge $(l_i, s_i)$ is deleted. Fig. 6 demonstrates this rearrangement.

**Lemma 1.** *DELETE preserves the RDT property.*

**Proof.** Let $T$ be the initial tree (Fig. 6a), $E = (s_i, s_{i+1}, \ldots, s_k)$ be an event of $T$, and $T'$ be the tree obtained from $T$ by applying DELETE to $E$ (Fig. 6b). Children of any node $s_j$ $(i \leq j \leq k)$ are denoted $l_j$ and $r_j$.

By definition, for any duplication history compatible with $T$ we have

$$l_i \prec l_{i+1} \prec \ldots \prec l_k \prec r_i \prec r_{i+1} \prec \ldots \prec r_k.$$

Thus, there is a way to partially agglomerate $T$ (using an RADT-like procedure) such that these nodes becomes leaves. The same agglomeration can be applied to $T'$ as only ancestors of the $l_j$s and $r_j$s are affected by DELETE. Now, 1) agglomerate the event $E$ of $T$, and 2) reduce $T'$ by agglomerating the cherry $(l_k, r_i)$ and then agglomerating the event $(s_{i+1}, \ldots, s_k)$. Two identical trees follow, which concludes the proof. □

By successively applying DELETE to any duplication tree, we remove all multiple duplication events. The following SPR rearrangement allows duplications to be moved within **simple** RDT, i.e., any RDT containing only simple duplications. Let $p$ be a node of a simple RDT $T$, $l$ its left child, $r$ its right child, and $x$ the left child of $r$. This rearrangement consists of pruning the subtree of root $x$ and regrafting it to the edge $(l, p)$ (Fig. 7). This rearrangement is an SPR (in fact an NNI); we name it **LEFT** as it moves the subtree root towards the left. It is obvious that the tree

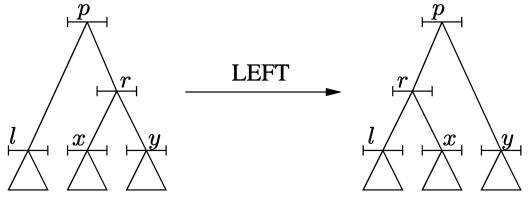



Fig. 7. LEFT rearrangement.

obtained by applying such a rearrangement to a simple RDT, is a simple RDT. We now establish the following lemma which shows that any simple tree can be transformed into a caterpillar.

**Lemma 2.** *Let $T$ be a simple RDT; $T$ can be transformed into a caterpillar by a succession of LEFT rearrangements.*

**Proof.** In a caterpillar all internal nodes are ancestors of 1. If $T$ is not a caterpillar, there is an internal node $r$ that is not an ancestor of 1. If $r$ is the right child of its father, we can apply LEFT to the left child of $r$ (Fig. 7). If $r$ is the left child of its father, we consider its father: It cannot be an ancestor of 1 since its children are $r$ and a node on the right of $r$. So, we can apply the same argument: Either the father of $r$ is adequate for performing LEFT, or we consider its father again. In this way, we necessarily obtain a node for which the rearrangement is possible. $T$ is then transformed into a caterpillar by successively applying the LEFT rearrangement to nodes which are not on the path between 1 and $\rho$. After a finite number of steps, all internal nodes are ancestors of 1 and $T$ has been transformed into a caterpillar. This concludes the proof of Lemma 2 and, therefore, of our Main Theorem. □

## 4 Local Search Method

We consider data consisting of an alignment of $n$ segments with length $k$, and of the ordering $O$ of the segments along the locus. This alignment has been created before tree construction and the problem is not to build simultaneously the alignment and the tree, a much more complicated task [29]. The aim is to find a (nearly) optimal duplication tree, where "optimal" is defined by some usual phylogenetic criterion and the ordered and aligned segments at hand. Topological rearrangements described in the previous section naturally lead to a local search method for this purpose. We discuss its use to optimize the usual Wagner parsimony [22] and the distance-based balanced minimum evolution criterion (BME) [30], [31]. First, we describe our local search method, then we define briefly these two criteria and explain how to compute them during local search.

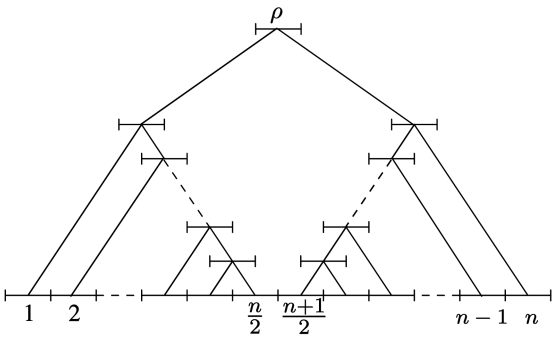



Fig. 8. A simple rooted duplication tree with a double caterpillar structure.

### 4.1 The LSDT Method

Our method, **LSDT (Local Search for Duplication Trees)**, follows a classical local search procedure in which, at each step, we try to strictly improve the current tree. This approach can be used to optimize various criteria. In this study, we restrict ourselves to parsimony and balanced minimum evolution; $f(T)$ represents the value (to be minimized) of one of these criteria for the duplication tree $T$ and the sequence set.

Algorithm 1
**Data**: An initial DT, $T_{best}$.
**Result**: A locally optimal DT for the selected criterion.
**while** $T_{best} \neq T_{current}$ **do**
  $T_{current} \leftarrow T_{best}$
  **foreach** tree $T_{new}$ obtained by applying an SPR to $T_{current}$ **do**
    **if** RADT($T_{new}$) = True **then**
      **if** $f(T_{new}) < f(T_{best})$ **then**
        replace $T_{best}$ by $T_{new}$
**return** $T_{best}$

Algorithm 1 summarizes LSDT. The neighborhood of the current DT, $T_{current}$, is computed using SPR. As we explained earlier, we use the RADT procedure to restrict this neighborhood to valid DTs. When a tree is a valid DT, its $f$ criterion value is computed. That way, we select the best neighbor of $T_{current}$. If this DT improves the value obtained so far (i.e., $f(T_{best})$), the local search restarts with this new topology. If no neighbor of $T_{current}$ improves $T_{best}$, the local search is stopped and returns $T_{best}$.

To analyze the time complexity of one LSDT step, we have to consider the size of the neighborhood defined by the restricted SPR. In the worst case, this size is of the same order as the size of an unrestricted SPR neighborhood, i.e., $O(n^2)$. Indeed for the "double caterpillar" (Fig. 8), it is possible to move any subtree being rooted on the path between $n/2$ and $\rho$ towards any edge of the path between $(n+1)/2$ and $\rho$; and inversely. Thus, for this tree, $O(n^2)$ restricted SPRs can be performed. In the worst case, restricting the neighborhood defined by SPR to duplication trees does not significantly decrease the neighborhood size. However, on average the diminution is quite significant; e.g., with $n = 48$, only 5 percent of the neighborhood corresponds to a valid DTs, assuming DTs are uniformly distributed [26].

Since the time complexity of the recognition algorithm (RADT) is $O(n)$, computing the neighborhood defined by restricted SPR requires $O(n^3)$. The calculation of the criterion value is done for each tree of the restricted neighborhood. Thus one local search step basically requires $O(n^3 + n^2 g)$, where $g$ represents the time complexity of computing the criterion value. However, preprocessing allows this time complexity to be lowered, both for parsimony and minimum evolution, as we shall explain in the following sections.

### 4.2 The Maximum Parsimony Criterion

Parsimony is commonly acknowledged [22] to be a good criterion when dealing with slightly divergent sequences, which is usually the case with tandemly duplicated genes [8]. The parsimony criterion involves selecting the tree which minimizes the number of substitutions needed to explain the evolution of the given sequences. Finding the most parsimonious tree [22] or duplication tree [15] is NP-hard, but we can find the optimal labeling of the internal nodes and the parsimony score of a given tree $T$ in polynomial time using the Fitch-Hartigan algorithm [32], [33]. The parsimony score and optimal labeling of internal nodes is independently computed for each position within sequences, using a **postorder depth-first search** algorithm that requires $O(n)$ time [32], [33]. Thus, computing the parsimony score of $n$ sequences of length $k$ requires $O(kn)$ time. Hence, if we use this algorithm during our local search method, one local search step is computed in $O(kn^3)$, which is relatively high.

To speed up this process, we adapted techniques commonly used in phylogeny for fast calculation of parsimony. Our implementation uses a data structure implemented (among others) in DNAPARS [24] and described in [34], [35]. Let $T_p$ be the pruned subtree and $T_r$ be the resulting tree. A preprocessing stage computes **the parsimony vector** (i.e., the optimal score and optimal labeling of all sequence positions) of every rooted subtree of $T_r$ using a double depth-first search [36] (Fig. 9a); the first search is postordered and computes the parsimony vector of down-subtrees; the second search is preordered and computes the parsimony vector of up-subtrees. Each search requires $O(nk)$ time. Thanks to this data structure, the parsimony score of the tree obtained by regrafting $T_p$ on any given edge of $T_r$ is computed in $O(k)$ (Fig. 9b). Hence, computing the SPR neighbor with minimum parsimony of any given duplication tree is achieved in $O(n^3 + n \times nk + n^2 k) = O(n^3 + n^2 k)$; the first term $(n^3)$ represents the neighborhood computation; the second term $(n \times nk)$ corresponds to the time required by the $n$

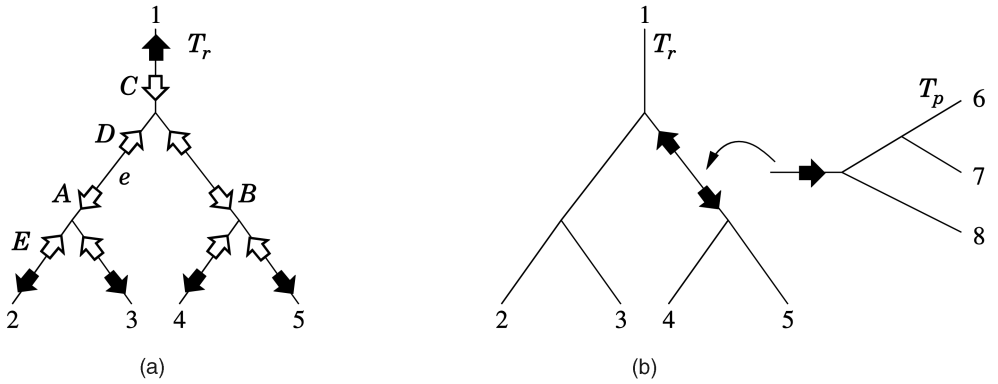



Fig. 9. (a) Every edge defines one down-subtree and one up-subtree; e.g., $A$ represents the down-subtree (2 3) defined by the edge $e$ while $D$ corresponds to the up-subtree (1 (4 5)). Moreover, only the parsimony vector of the five leaves is known before the preprocessing stage. The postorder search computes the parsimony vector of down-subtrees: $A$ is computed from 2 and 3, $B$ from 4 and 5, $C$ from $A$ and $B$. The preorder search computes the parsimony vector of up-subtrees: $D$ is obtained from 1 and $B$, $E$ is obtained from $D$ and 3, etc. (b) When the parsimony vector of every subtree in $T_r$ is known, regrafting $T_p$ on any given edge and computing the parsimony score of the resulting tree only requires analyzing the parsimony vector of three subtrees and is done in $O(k)$ time.

preprocessing stages; the third term $(n^2k)$ is the time to test the $n$ subtrees and the $n$ possible insertion edges.

### 4.3 The Distance-Based Balanced Minimum Evolution Principle

As in any distance-based approach, we first estimate the matrix of pairwise evolutionary distances between the segments, using some standard distance estimator [22], e.g., the Kimura two-parameter estimator [37] in case of DNA or the JTT method with proteins [38]. Let $\Delta$ be this matrix and $\delta_{ij}$ be the distance between segments $i$ and $j$. The $\Delta$ matrix plus the segment order is the input of the reconstruction method.

The **minimum evolution principle (ME)** [39], [40] involves selecting the shortest tree to be the tree which best explains the observed sequences. The tree length is equal to the sum of all the edge lengths, and the edge lengths are estimated by minimizing a least squares fit criterion. The problem of inferring optimal phylogenies within ME is commonly assumed to be NP-hard, as are many other distance-based phylogeny inference problems [41]. Nonetheless, ME forms the basis of several phylogenetic reconstruction methods, generally based on greedy heuristics. Among them is the popular Neighbor-Joining (NJ) algorithm [17]. Starting from a star tree, NJ iteratively agglomerates external pairs of taxa so as to minimize the tree length at each step.

Recently, Pauplin [30] proposed a new simple formula to estimate the tree length $L(T)$ of tree $T$:

$$L(T) = \sum_{i<j} 2^{1-\mathcal{T}_{ij}}\delta_{ij},$$

where $\mathcal{T}_{ij}$ is the topological distance (number of edges) in $T$ between segments $i$ and $j$. The correctness of this formula was shown by Semple and Steel [42], while Desper and Gascuel [31] showed that this formula is a special case of weighted-least squares tree fitting. Moreover, Desper and Gascuel demonstrated that selecting the shortest tree (as computed from above formula) is statistically consistent and well suited for phylogenetic inference. They called this new version of ME **"balanced minimum evolution" (BME)** [31].

Using the above formula, the length of any given tree is computed in $O(n^2)$, so computing one LSDT local search step can be achieved in $O(n^4)$. However, a faster implementation is possible using a straightforward modification of our BME addition algorithm [43]. This involves:

1. pruning a rooted subtree $T_p$ from tree $T$,
2. computing the average distance between all non-intersecting subtree pairs in the remaining tree $T_r$,
3. computing the average distance between $T_p$ and any subtree of $T_r$ in $T$, and
4. using formula (10) from [43] and RADT to find the best allowed edge to regraft $T_p$.

Steps 2 and 3 are based on algorithms described in [43], which follow the same approach as the double depth-first search described in the previous section. These two steps require $O(n^2)$, just as Step 4. As there are $O(n)$ subtrees to prune and regraft, this implementation requires $O(n^3)$ to perform one search step.

## 5 Results

### 5.1 Simulation Protocol

We applied our method and other existing methods to simulated datasets obtained using the procedure described in [18]. We uniformly randomly generated rooted tandem duplication trees (see [26]) with 12, 24, and 48 leaves and assigned lengths to the edges of these trees using the coalescent model [44]. We then obtained **molecular clock trees (MC)**, which might be unrealistic in numerous cases, e.g., when the sequences being studied contain pseudogenes which evolve much faster than functional genes. Then, we generated **nonmolecular clock trees (NO-MC)** from the previous trees by independently multiplying

every edge length by $1 + 0.8X$, where $X$ was drawn from an exponential distribution with parameter 1. MC trees were rescaled by multiplying every edge length by 1.8. The trees thus obtained (MC and NO-MC) have a maximum leaf-to-leaf divergence in the range $[0.1, 0.7]$, and in NO-MC trees the ratio between the longest and shortest root-to-leaf lineages is about 3.0 on average. Both values are in accordance with real data, e.g., gene families [8] or repeated protein domains [10].

SEQGEN [45] was used to produce a 1,000 bp-long nucleotide multiple alignment from each of the generated trees using the Kimura two-parameter model of substitution [46], and a distance matrix was computed by DNADIST [24] from this alignment using the same substitution model. For MC and NO-MC cases, 1,000 trees (and, then, 1,000 sequence sets and 1,000 distance matrices) were generated per tree size. These data sets were used to compare the ability of the various methods to recover the original trees from the sequences or from the distance matrices, depending on the method being tested. We measured the percentage of trees (out of 1,000) being correctly reconstructed ($\%tr$). For the phylogeny reconstruction methods, we also kept the percentage of duplication trees among the set of inferred trees. Due to the random process used for generating these trees and datasets, some short branches might not have undergone any substitution (as during Evolution) and, thus, are unobtainable, except by chance. When $n$ and, thus, the branch number is high, it becomes hard or impossible to find the entire tree. So, we also measured the percentage of duplication events in the true tree recovered by the inferred tree ($\%ev$). A duplication event involves one or more internal nodes and is the lowest common ancestor of a set of leaves; we say it "covers" its descendent leaves. However, the leaves covered by a simple duplication event can change when the root position changes. As regards the true tree, the root is known and each event is defined by the set of leaves which it covers. But, the inferred tree is unrooted. To avoid ambiguity, we then tested all possible root positions and chose the one which gave the highest proximity in number of events detected between the true tree and the inferred tree, where two events are identical if they cover the same leaves. Finally, we kept the average parsimony value of each method ($pars$).

### 5.2 Performance and Comparison

Using this protocol, we compared NJ [17], TNT [47], and GREEDY-SEARCH (GS) [21] which starts from the NJ tree, a modified version of GREEDY TRHIST RESTRICTED (GTR) [9] to infer multiple duplication trees, WINDOWS [10], DTSCORE [18], and eight versions of our local search method LSDT corresponding to different starting duplication trees (GS, GTR, WINDOW, and DTSCORE) and different criteria (parsimony and BME). TNT and GS use the parsimony criterion, but the other are distance-based methods. TNT is acknowledged as one of the very best parsimony packages; it was run with 10 replicates and TBR rearrangements. TNT often returns a set of equally parsimonious trees. When this set contained duplication trees, we randomly selected one of them; when no duplication tree was inferred by TNT, we randomly selected one of the output trees.

Results are given in Tables 1 and 2. First, we observe that with $n = 48$ the true tree is almost never entirely found, for the reasons explained earlier. On the other hand, the best methods recover 80 to 95 percent of the duplication events, indicating that the tested datasets are relatively easy. NJ and TNT perform relatively well, but they often output trees that are not duplication trees, which is unsatisfactory (e.g., with 48 leaves and NO-MC, NJ and TNT only infer 1 percent and 5 percent of duplication trees, respectively). The GS approach is noteworthy since it modifies the trees inferred by NJ to transform them into duplication trees. However, GS is only slightly better than NJ regarding the proportion of correctly reconstructed trees, but considerably degrades the number of recovered duplication events, which could be explained by the blind search it performs to transform NJ trees into duplication trees. GTR also obtains relatively poor results. As expected from its assumptions, WINDOW performs better in the MC case than in the NO-MC one. Finally, DTSCORE obtains the best performance among the four existing methods, whatever the topological criterion considered.

Applying our method to starting trees produced by GS, GTR, WINDOW, and DTSCORE reveals the advantages of the local search approach. Optimizing parsimony or BME gives similar results, with a slight advantage for parsimony as expected from the relatively low divergence rates in our data sets. The trees produced by GS, GTR, and WINDOW are clearly improved and, for most, are better than those obtained by DTSCORE. DTSCORE trees are also improved, even though this improvement is not very high from a topological point of view. This could be explained by the fact that DTSCORE is already an accurate method with respect to the datasets used.

When we consider the parsimony criterion, the gain achieved by LSDT is appreciable for each start method. This could be expected for GS, WINDOW and DTSCORE which do not optimize this criterion; with $n = 48$ in NO-MC case, the gain for GS is about 329, thus confirming that this method is clearly suboptimal; the gains for WINDOW and DTSCORE are about 42 and 15, which are lower but still significant. The GTR results, which optimizes parsimony, are more surprising since the gain (always with $n = 48$ in NO-MC case) is about 77 on average, which is very high. Moreover, the parsimony value obtained by LSDT is very close to that of TNT, in spite of a much more restricted search space. This confirms the good performance of our

TABLE 1
Performance Comparison Using Simulations (Molecular Clock Mode of Evolution)

| | 12 leaves | | | 24 leaves | | | 48 leaves | | |
|---|---|---|---|---|---|---|---|---|---|
| | % $tr$ | % $ev$ | $pars$ | % $tr$ | % $ev$ | $pars$ | % $tr$ | % $ev$ | $pars$ |
| NJ | 50.2 (70) | 92.4 | – | 13.4 (29) | 88.2 | – | 0.2 (1) | 82.0 | – |
| TNT | 58.4 (80) | 94.0 | 443 | 21.4 (42) | 90.2 | 733 | 1.4 (6) | 84.1 | 1136 |
| GS | 51.6 | 87.2 | 467 | 14.8 | 68.8 | 833 | 0.2 | 48.8 | 1453 |
| GS+LSDT_MP | 67.6 | 94.8 | −23 | 33.0 | 91.6 | −94 | 3.4 | 84.3 | −292 |
| GS+LSDT_BME | 63.4 | 94.4 | – | 32.0 | 92.1 | – | 3.4 | 86.9 | – |
| GTR | 33.6 | 87.2 | 450 | 3.4 | 77.5 | 759 | 0.0 | 68.5 | 1204 |
| GTR+LSDT_MP | 71.6 | 96.0 | −6 | 36.6 | 93.1 | −24 | 4.2 | 88.3 | −59 |
| GTR+LSDT_BME | 64.6 | 95.0 | – | 33.2 | 92.8 | – | 3.0 | 88.6 | – |
| WINDOW | 52.0 | 92.8 | 444 | 13.4 | 87.9 | 738 | 0.4 | 81.5 | 1149 |
| WINDOW+LSDT_MP | 70.0 | 96.0 | −1 | 38.0 | 93.8 | −4 | 4.6 | 90.1 | −10 |
| WINDOW+LSDT_BME | 64.6 | 95.0 | – | 33.8 | 93.3 | – | 4.6 | 89.7 | – |
| DTSCORE | 61.8 | 94.4 | 445 | 31.2 | 91.2 | 738 | 2.4 | 85.6 | 1149 |
| DTSCORE+LSDT_MP | 70.2 | 96.0 | −2 | 37.8 | 93.5 | −4 | 5.4 | 89.9 | −10 |
| DTSCORE+LSDT_BME | 65.0 | 95.1 | – | 33.8 | 93.1 | – | 5.2 | 89.8 | – |

*X+LSDT_Y: X is the method used to obtain the starting tree and Y the criterion being optimized by LSDT; $\%tr$: the percentage of trees being correctly reconstructed; the percentage of duplication trees obtained by phylogeny reconstruction methods is given between parentheses; $\%ev$: the percentage of duplication events in the true tree being recovered by the inferred tree; $pars$: the average parsimony value.*

local search method. It should be stressed that these gains are obtained at low computational cost as dealing with any of the 48-taxon datasets only requires about 10 seconds for parsimony and five seconds for BME on a standard PC-Pentium 4.

### 5.3 Analysis of the ZNF45 Family

Zinc finger (ZNF) genes code for proteins that contain one or more zinc finger motifs. The zinc finger motif is one of the most common motifs involved in nucleic acid-protein interaction. Experimental studies on functions of ZNF genes suggest that many of them code for transcription factors, and some of them are known to take part in cellular growth and development [48]. However, the biological functions of most ZNF genes are currently unknown. The 16 members of ZNF45 gene family are found in the q13.2 gene cluster on human chromosome 19 [49]. The organization and features of the members of the ZNF45 family suggest that the genes in the family may have been produced by a series of *in situ*

TABLE 2
Performance Comparison Using Simulations (No Molecular Clock of Evolution)

| | 12 leaves | | | 24 leaves | | | 48 leaves | | |
|---|---|---|---|---|---|---|---|---|---|
| | % $tr$ | % $ev$ | $pars$ | % $tr$ | % $ev$ | $pars$ | % $tr$ | % $ev$ | $pars$ |
| NJ | 44.6 (63) | 90.8 | – | 9.0 (22) | 86.0 | – | 0.0 (1) | 79.7 | – |
| TNT | 52.4 (75) | 92.6 | 439 | 15.6 (37) | 88.4 | 715 | 0.4 (5) | 81.7 | 1135 |
| GS | 46.8 | 82.8 | 464 | 9.6 | 65.3 | 833 | 0.0 | 46.0 | 1493 |
| GS+LSDT_MP | 61.9 | 93.5 | −23 | 26.6 | 89.4 | −110 | 1.9 | 82.2 | −329 |
| GS+LSDT_BME | 59.7 | 93.5 | – | 26.7 | 90.7 | – | 2.9 | 85.2 | – |
| GTR | 25.7 | 82.9 | 449 | 2.2 | 71.9 | 749 | 0.0 | 61.8 | 1223 |
| GTR+LSDT_MP | 65.0 | 94.9 | −9 | 28.5 | 91.2 | −31 | 2.1 | 85.7 | −77 |
| GTR+LSDT_BME | 60.3 | 94.0 | – | 26.5 | 91.3 | – | 2.5 | 86.7 | – |
| WINDOW | 26.4 | 85.5 | 445 | 3.3 | 75.8 | 736 | 0.0 | 67.2 | 1186 |
| WINDOW+LSDT_MP | 65.1 | 95.0 | −6 | 25.4 | 91.5 | −19 | 1.7 | 86.1 | −42 |
| WINDOW+LSDT_BME | 60.1 | 93.9 | – | 26.3 | 91.3 | – | 2.7 | 86.9 | – |
| DTSCORE | 55.2 | 92.5 | 442 | 18.8 | 89.0 | 722 | 0.7 | 83.6 | 1155 |
| DTSCORE+LSDT_MP | 62.6 | 94.5 | −2 | 28.9 | 92.3 | −6 | 2.5 | 88.5 | −15 |
| DTSCORE+LSDT_BME | 60.9 | 93.9 | – | 28.2 | 91.9 | – | 2.8 | 88.1 | – |

*Note: see Table 1.*

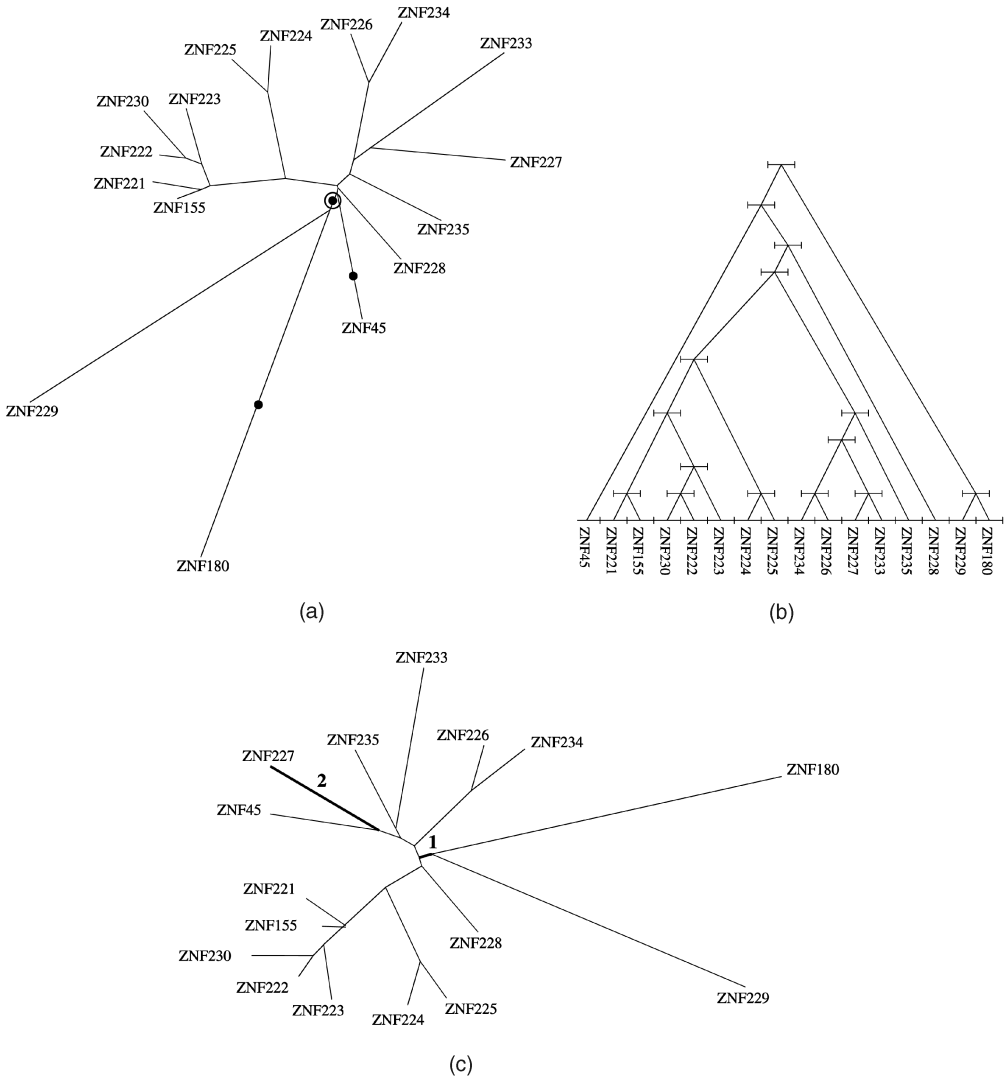



Fig. 10. (a) Duplication tree for the 16 genes of human ZNF 45 family inferred by DTSCORE plus LSDT with parsimony; black dots represent the only allowed root positions, according to the tandem duplication model; the (arbitrarily) selected root position is circled. (b) Rooted duplication tree corresponding to tree (a). (c) Phylogeny inferred by TNT. Tree (a) can be obtained from tree (c) by moving ZNF45 and ZNF228 to edge 1, and ZNF233 to edge 2. Edge lengths in tree (a) and tree (c) were estimated by maximum likelihood [52]. Lengths in tree (b) are meaningless and were adjusted to obtain a readable drawing.

gene duplication events [49]. The ZNF45 gene family has been previously studied by Tang et al. [10] and Zhang et al. [21], who proposed different tandem duplication trees to explain its evolutionary history.

We downloaded the DNA sequences of the 16 members of ZNF45 from NCBI. Multiple alignment was achieved using TCOFFEE,[1] using default settings. We removed gaps as usual in phylogenetics [22] and third codon positions which look saturated (734 parsimony steps are required to explain the evolution of the 237 sites). We thus obtained a final alignment[2] containing 474 homologous sites, with a maximum pairwise divergence of 0.45.

PAUP* [23] was used to estimate the matrix of pairwise distances, assuming the GTR substitution model [50] and a gamma distribution of rates with parameter 1.

1. http://igs-server.cnrs-mrs.fr/Tcoffee/tcoffee_cgi/index.cgi.
2. Available on request.

We used this distance matrix and DTSCORE to build a starting tree, which was then refined by LSDT using parsimony. We selected this criterion because of its good performance with simulated data (Tables 1 and 2). The resulting tree (Figs. 10a and 10b) is a simple DT requiring 897 steps to explain the extant sequences. We tried to improve this score using a computationally intensive ratchet approach [51], but were unable to obtain any other DT with better (or even identical) parsimony. We also ran TNT with ratchet, 1,000 random taxon addition replicates and TBR branch swapping (i.e., all TNT options to intensify the search) and found one maximum-parsimony phylogeny requiring 896 steps. This phylogeny (Fig. 10c) contains an unresolved node with degree 4 and is not a duplication tree.

TNT phylogeny is close to LSDT duplication tree. To transform from one to the other only three taxa have to be

TABLE 3
Analysis of the ZNF45 Data Set

| | # Parsimony score | % events shared with the LSDT tree of Fig. 10a | # Parsimony score after using LSDT |
|---|---|---|---|
| Tang *et al.* [10] | 926 | 46 | 897 (−29) |
| Zhang *et al.* [21] | 923 | 60 | 897 (−26) |
| TNT | 896 | 67 | – |
| GS | 947 | 53 | 897 (−50) |
| GTR | 911 | 60 | 901 (−10) |
| WINDOW | 907 | 67 | 897 (−10) |
| DTSCORE | 915 | 73 | 897 (−18) |

moved (Fig. 10), and both trees differ by only 1 parsimony step. A similar difference was commonly observed in simulation where TNT found (non-DT) phylogenies requiring one parsimony step less (on average) than the DTs found by LSDT (Tables 1 and 2), though the true tree used to generate the sequences was a DT. Thus, having (only) one parsimony step of difference between the best DT and the best phylogeny is not significant and can be seen as supporting the duplication model. Moreover, the discrepancy between the two trees can be explained by long branch attraction, a phenomenon that frequently affects parsimony-based reconstructions [53]. Indeed, ZNF180 and ZNF229 genes are distant from the other genes (Figs. 10a and 10c) and might perturb the whole tree. When removing those two genes from the data set, both LSDT and TNT found the same tree, which is identical to the LSDT tree of Fig. 10a without the two genes. With 14 segments, the probability of randomly picking up a duplication tree among all distinct phylogenies is less than $10^{-4}$ [26]. This extremely small probability indicates that the identity between LSDT and TNT trees is very unlikely to be due to chance. This provides a strong support for the tandem duplication model and indicates that our LSDT tree likely represents most—if not all—of the history of ZNF45 family.

We compared trees obtained by Tang et al. [10], Zhang et al. [21], and those of the other programs to the LSDT tree of Fig. 10. We computed the parsimony score of each tree and the percentage of events shared by each tree with the LSDT tree. Just as in the simulation study, we tested GS [21], GTR [9], WINDOW [10], DTSCORE [8], and LSDT using different starting points but optimizing parsimony in all cases.

Results are displayed in Table 3 and confirm those obtained with simulated data sets.Results of trees from [10] and [21] are poor, which was expected as these methods (WINDOWS and GS, respectively) do not optimize the parsimony criterion and as we did not use the same alignment. GS is relatively poor, while DTSCORE, WINDOWS, and GTR perform better. LSDT clearly improves these four methods, with gains ranging from 10 to 50 parsimony steps. In all cases but GTR, LSDT recovers the most parsimonious DT of Fig. 10.

## 6 Conclusion and Prospects

We have demonstrated that restricting the neighborhood defined by the SPR rearrangement to valid duplication trees allows the whole DT space to be explored. Thanks to these rearrangements, we have defined a general local search method which we used to optimize the parsimony and balanced minimum evolution criteria. We have thus improved the topological accuracy of all the tested methods.

Several research directions are possible. Finding the set of combinatorial configurations for the SPR rearrangement which necessarily produce a duplication tree, could allow the neighborhood computation to be accelerated (e.g., for $n = 48$ only 5 percent of the SPR neighborhood correspond to duplication trees) and, furthermore, gain more insight into the nature of duplication trees, which are just starting to be investigated mathematically [12], [26], [27]. Our local search method could be improved using restricted TBR rearrangements or with the help of different stochastic approaches (taboo, noising, ...) in order to avoid local minima. Moreover, it would be relevant to test this local search method with other criteria like maximum likelihood. Finally, combining the tandem duplication events with speciation events, as described in [54] and [55] for nontandem duplications, would be relevant for real applications where we have homologous tandem repeats from several genomes.

## Acknowledgments

The authors would like to thank Wafae El Alaoui for her help with ZNF45 family genes, and Richard Desper, Wim Hordijk and the referees of the Workshop on Algorithms in Bioinformatics (WABI '04) for reading preliminary versions of this paper. This work was supported by ACI-IMPBIO (Ministère de la Recherche, France).

## REFERENCES


[1] F. Blattner, G. Plunkett, C. Bloch, N. Perna, V. Burland, M. Riley, J. Collado-Vides, J. Glasner, C. Rode, G. Mayhew, J. Gregor, N. Davis, H. Kirkpatrick, M. Goeden, D. Rose, B. Mau, and Y. Shao, "The Complete Genome Sequence Of Escherichia Coli k-12," *Science,* vol. 277, no. 5331, pp. 1453-1474, 1997.

[2] E. Lander et al., "Initial Sequencing and Analysis of the Human Genome," *Nature,* vol. 409, pp. 860-921, 2001.

[3] A. Smit, "Interspersed Repeats and Other Mementos of Transposable Elements in Mammalian Genomes," *Current Opinion in Genetics & Development,* vol. 9, pp. 657-663, 1999.

[4] W. Fitch, "Phylogenies Constrained by Cross-Over Process as Illustrated by Human Hemoglobins in a Thirteen-Cycle, Eleven Amino-Acid Repeat in Human Apolipoprotein A-I," *Genetics,* vol. 86, pp. 623-644, 1977.

[5] G. Levinson and G. Gutman, "Slipped-Strand Mispairing: A Major Mechanism for DNA Sequence Evolution," *Molecular Biology and Evolution,* vol. 4, pp. 203-221, 1987.

[6] J. Zhang and M. Nei, "Evolution of Antennapedia-Class Homeobox Genes," *Genetics,* vol. 142, no. 1, pp. 295-303, 1996.

[7] O. Elemento and O. Gascuel, "An Exact and Polynomial Distance-Based Algorithm to Reconstruct Single Copy Tandem Duplication Trees," *Proc. 14th Ann. Symp. Combinatorial Pattern Matching (CPM2003),* 2003.

[8] O. Elemento, O. Gascuel, and M.-P. Lefranc, "Reconstructing the Duplication History of Tandemly Repeated Genes," *Molecular Biology and Evolution,* vol. 19, pp. 278-288, 2002.

[9] G. Benson and L. Dong, "Reconstructing the Duplication History of a Tandem Repeat," *Proc. Intelligent Systems in Molecular Biology (ISMB1999),* T. Lengauer, ed., pp. 44-53, 1999.

[10] M. Tang, M. Waterman, and S. Yooseph, "Zinc Finger Gene Clusters and Tandem Gene Duplication," *J. Computational Biology,* vol. 9, pp. 429-446, 2002.

[11] E. Rivals, "A Survey on Algorithmic Aspects of Tandem Repeats Evolution," *Int'l J. Foundations of Computer Science,* vol. 15, no. 2, pp. 225-257, 2004.

[12] O. Gascuel, D. Bertrand, and O. Elemento, "Reconstructing the Duplication History of Tandemly Repeated Sequences," *Math. of Evolution and Phylogeny,* O. Gascuel, ed., 2004.

[13] S. Ohno, *Evolution by Gene Duplication.* Springer Verlag, 1970.

[14] P.L. Fleche, Y. Hauck, L. Onteniente, A. Prieur, F. Denoeud, V. Ramisse, P. Sylvestre, G. Benson, F. Ramisse, and G. Vergnaud, "A Tandem Repeats Database for Bacterial Genomes: Application to the Genotyping of *Yersinia Pestis* and *Bacillus Anthracis*," *BioMed Central Microbiology,* vol. 1, pp. 2-15, 2001.

[15] D. Jaitly, P. Kearney, G. Lin, and B. Ma, "Methods for Reconstructing the History of Tandem Repeats and Their Application to the Human Genome," *J. Computer and System Sciences,* vol. 65, pp. 494-507, 2002.

[16] P. Sneath and R. Sokal, *Numerical Taxonomy.* pp. 230-234, San Francisco: W.H. Freeman and Company, 1973.

[17] N. Saitou and M. Nei, "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees," *Molecular Biology and Evolution,* vol. 4, pp. 406-425, 1987.

[18] O. Elemento and O. Gascuel, "A Fast and Accurate Distance-Based Algorithm to Reconstruct Tandem Duplication Trees," *Bioinformatics,* vol. 18, pp. 92-99, 2002.

[19] J. Barthélemy and A. Guénoche, *Trees and Proximity Representations.* Wiley and Sons, 1991.

[20] S. Sattath and A. Tversky, "Additive Similarity Trees," *Psychometrika,* vol. 42, pp. 319-345, 1977.

[21] L. Zhang, B. Ma, L. Wang, and Y. Xu, "Greedy Method for Inferring Tandem Duplication History," *Bioinformatics,* vol. 19, pp. 1497-1504, 2003.

[22] D. Swofford, P. Olsen, P. Waddell, and D. Hillis, *Molecular Systematics.* pp. 407-514, Sunderland, Mass.: Sinauer Associates, 1996.

[23] D. Swofford, *PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods),* version 4. Sunderland, Mass.: Sinauer Associates, 1999.

[24] J. Felsenstein, "PHYLIP—PHYLogeny Inference Package," *Cladistics,* vol. 5, pp. 164-166, 1989.

[25] C. Semple and M. Steel, *Phylogenetics.* Oxford Univ. Press, 2003.

[26] O. Gascuel, M. Hendy, A. Jean-Marie, and S. McLachlan, "The Combinatorics of Tandem Duplication Trees," *Systematic Biology,* vol. 52, pp. 110-118, 2003.

[27] J. Yang and L. Zhang, "On Counting Tandem Duplication Trees," *Molecular Biology and Evolution,* vol. 21, pp. 1160-1163, 2004.

[28] D. Robinson, "Comparison of Labeled Trees with Valency Trees," *J. Combinatorial Theory,* vol. 11, pp. 105-119, 1971.

[29] L. Wang and D. Gusfield, "Improved Approximation Algorithms for Tree Alignment," *J. Algorithms,* vol. 25, pp. 255-273, 1997.

[30] Y. Pauplin, "Direct Calculation of a Tree Length Using a Distance Matrix," *J. Molecular Evolution,* vol. 51, pp. 41-47, 2000.

[31] R. Desper and O. Gascuel, "Theoretical Foundation of the Balanced Minimum Evolution Method of Phylogenetic Inference and Its Relationship to Weighted Least-Squares Tree Fitting," *Molecular Biology and Evolution,* vol. 21, no. 3, pp. 587-598, 2004.

[32] W. Fitch, "Toward Defining the Course of Evolution: Minimum Change for a Specified Tree Topology," *Systematic Zoology,* vol. 20, pp. 406-416, 1971.

[33] J. Hartigan, "Minimum Mutation Fits to a Given Tree," *Biometrics,* vol. 29, pp. 53-65, 1973.

[34] G. Ganapathy, V. Ramachandran, and T. Warnow, "Better Hill-Climbing Searches for Parsimony," *Proc. Third Int'l Workshop Algorithms in Bioinformatics,* 2003.

[35] P.A. Goloboff, "Methods for Faster Parsimony Analysis," *Cladistics,* vol. 12, pp. 199-220, 1996.

[36] V. Berry and O. Gascuel, "Inferring Evolutionary Trees with Strong Combinatorial Evidence," *Theoretical Computer Science,* vol. 240, pp. 271-298, 2000.

[37] M. Kimura, "A Simple Model for Estimating Evolutionary Rates of Base Substitutions through Comparative Studies of Nucleotide Sequences," *J. Molecular Evolution,* vol. 16, pp. 111-120, 1980.

[38] D. Jones, W. Taylor, and J. Thornton, "The Rapid Generation of Mutation Data Matrices from Protein Sequences," *Computer Applications in Biosciences,* vol. 8, pp. 275-282, 1992.

[39] K. Kidd and L. Sgaramella-Zonta, "Phylogenetic Analysis: Concepts and Methods," *Am. J. Human Genetics,* vol. 23, pp. 235-252, 1971.

[40] A. Rzhetsky and M. Nei, "Theoretical Foundation of the Minimum-Evolution Method of Phylogenetic Inference," *Molecular Biology and Evolution,* vol. 10, pp. 173-1095, 1993.

[41] W. Day, "Computational Complexity of Inferring Phylogenies from Dissimilarity Matrices," *Bull. Math. Biology,* vol. 49, pp. 461-467, 1987.

[42] C. Semple and M. Steel, "Cyclic Permutations and Evolutionary Trees," *Advances in Applied Math.,* vol. 32, no. 4, pp. 669-680, 2004.

[43] R. Desper and O. Gascuel, "Fast and Accurate Phylogeny Reconstruction Algorithms Based on the Minimum-Evolution Principle," *J. Computational Biology,* vol. 9, pp. 687-706, 2002.

[44] M. Kuhner and J. Felsenstein, "A Simulation Comparison of Phylogeny Algorithms under Equal and Unequal Evolutionary Rates," *Molecular Biology and Evolution,* vol. 11, pp. 459-468, 1994.

[45] A. Rambault and N. Grassly, "Seq-Gen: An Application for the Monte Carlo Simulation of DNA Sequence Evolution Along Phylogenetic Trees," *Computer Applied Biosciences,* vol. 13, pp. 235-238, 1997.

[46] J. Felsenstein and G. Churchill, "A Hidden Markov Model Approach to Variation Among Sites in Rate of Evolution," *Molecular Biology and Evolution,* vol. 13, pp. 93-104, 1996.

[47] P.A. Goloboff, J.S. Farris, and K. Nixon, "TNT: Tree Analysis Using New Technology," 2000, www.cladistics.com.

[48] T. El-Barabi and T. Pieler, "Zinc Finger Proteins: What We Know and What We Would Like to Know," *Mechanisms of Development,* vol. 33, pp. 155-169, 1991.

[49] M. Shannon, J. Kim, L. Ashworth, E. Branscomb, and L. Stubbs, "Tandem Zinc-Finger Gene Families in Mammals: Insights and Unanswered Questions," *DNA Sequence—The J. Sequencing and Mapping,* vol. 8, no. 5, pp. 303-315, 1998.

[50] P. Waddel and M. Steel, "General Time Reversible Distances with Unequal Rates Across Sites: Mixing T and Inverse Gaussian Distributions with Invariant Sites," *Molecular Phylogeny and Evolution,* vol. 8, pp. 398-414, 1997.

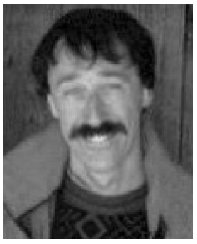
[51] K.C. Nixon, "The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis," *Cladistics,* vol. 15, pp. 407-414, 1999.

[52] S. Guindon and O. Gascuel, "A Simple, Fast and Accurate Method to Estimate Large Phylogenies by Maximum-Likelihood," *Systematic Biology,* vol. 52, no. 5, pp. 696-704, 2003.

[53] J. Felsenstein, "Cases in Which Parsimony or Compatibility Methods Will Be Positively Misleading," *Systematic Zoology,* vol. 27, pp. 401-410, 1978.

[54] D. Page and M. Charleston, "From Gene to Organismal Phylogeny: Reconciled Trees and the Gene Tree/Species Tree Problem," *Molecular Phylogenetics and Evolution*, vol. 7, pp. 231-240, 1997.
[55] M. Hallett, J. Lagergren, and A. Tofigh, "Simultaneous Identification of Duplications and Lateral Transfers," *Proc. Conf. Research and Computational Molecular Biology (RECOMB2004)*, pp. 347-356, 2004.

**Denis Bertrand** is a PhD student under the supervision of Olivier Gascuel. His research subject is the study of tandemly repeated sequences. His main areas of interest are phylogenetics, combinatorics, and algorithms.



**Olivier Gascuel** is Directeur de Recherche at the Centre National de la Recherche Scientifique (France). He is the head of the bioinformatics group from the LIRMM laboratory, belongs to the editorial board of Systematic Biology and of BMC Evolutionary Biology, and has served in a number of program committees of bioinformatics conferences (ISMB, WABI). He started in this field in the mid 1980s, with works on sequence analysis and protein structure prediction. Since the beginning of the 1990s, he turned his efforts to phylogenetics, focusing on the mathematical and computational tools and concepts. He (co)authored several well-known phylogeny inference programs (BioNJ, PHYML, FastME).