Reconstructing Ancestral Gene Orders Using Conserved Intervals

Anne Bergeron, Annie Chateau, Cedric Chauve
LaCIM
Université du Québec à Montréal

Mathieu Blanchette
McGill Center for Bioinformatics and School of Computer Science, McGill University

Proceedings of Workshop on Algorithms in Bioinformatics 2004 (WABI'04), volume 3240 of Lecture Notes in Bioinformatics, p. 14-25.

Reconstructing Ancestral Gene Orders Using Conserved Intervals

Anne Bergeron[1], Mathieu Blanchette[2], Annie Chateau[1], and Cedric Chauve[1]

[1] LaCIM, Université du Québec à Montreal, Canada.
[2] McGill Center for Bioinformatics, Canada

Abstract. Conserved intervals were recently introduced as a measure of similarity between genomes whose genes have been shuffled during evolution by genomic rearrangements. Phylogenetic reconstruction based on such similarity measures raises many biological, formal and algorithmic questions, in particular the labelling of internal nodes with putative ancestral gene orders, and the selection of a good tree topology. In this paper, we investigate the properties of sets of permutations associated to conserved intervals as a representation of putative ancestral gene orders for a given tree topology. We define set-theoretic operations on sets of conserved intervals, together with the associated algorithms, and we apply these techniques, in a manner similar to the Fitch-Hartigan algorithm for parsimony, to a subset of chloroplast genes of 13 species.

1 Introduction

The information contained in the order in which genes occur on the genomes of different species has proved very useful for inferring phylogenetic relationships (see [18] for a review). Together with phylogenetic information, ancestral gene order reconstructions give some clues about the conservation of the functional organisation of genomes, towards a global knowledge of life evolution. With a few exceptions [16], phylogeny reconstruction techniques using gene order data rely on the definition of an evolutionary distance between two gene orders. These distances are usually computed as the minimal number of rearrangement operations needed to transform one genome into another, for a *fixed* set of rearrangement operations. Since most choices lead quickly to hard problems, the set of operations is usually restricted to reversals, translocations, fusions or fissions, in which case a linear-time algorithm exists ([1, 13, 14] and [3] for a review). However, this choice of rearrangement operations is more dictated by algorithm necessity than by biological reality, as rearrangements such as transpositions and inverted transpositions could be quite common in some genomes (see [6] for heuristics dealing with these types of rearrangements).

A family of phylogenetic approaches dubbed "distance-based" methods only rely on the ability to compute pair-wise evolutionary distances between extant species, which are then fed into an algorithm such as neighbor-joining (see [11] for a review) to infer a tree topology and branch lengths for the species considered. While these approaches have proved very useful for phylogenetic inference

[22], they provide information neither about the putative ancestral gene orders nor about the evolutionary process that led to the extant species. In contrast, parsimony-based approaches attempt to identify the rearrangement scenario (including tree topology and gene orders at the internal nodes) that minimizes the number of evolutionary events required. This formulation usually leads to much more difficult computational problems [9], although good heuristics have been developed for breakpoint [5, 19, 21] and reversal [8, 17, 23] distances. It provides a candidate explanation, in terms of ancestral gene orders and rearrangements applied on them, for the modern gene orders. However, these methods only provide us with one (or a small number of) possible hypothesis about ancestral gene orders, with no information about alternate optimal or near-optimal solutions.

In this study, we develop the mathematical tools and algorithms required to describe and infer a set of likely ancestral gene orders at each internal node of a phylogenetic tree with a *given* topology. We use the notion of *conserved intervals*, introduced in [4], as a measure of similarity for sets of permutations representing genomes with equal gene contents. In short, a conserved interval is a generalization of the notion of gene adjacency, corresponding to a constraint on the ordering of the genes. This type of representation has several properties that make it is particularly useful in the study of gene order: (i) it is a compact representation of a rich set of gene orders (e.g. putative ancestral gene orders), (ii) it provides computationally tractable operations on these sets (some originally described in [4], others reported here), (iii) it is intimately related to the reversal distance computation [3], although it behaves well even in the presence of other types of intra-chromosomal rearrangements like transpositions and inverted transposition, and (iv) it is particularly effective at dealing with short rearrangement events, which seem to be the most common in mitochondrial and chloroplastic genomes [20].

In Section 2, we introduce the notion of conserved intervals and illustrate it using a small example. Section 3 reviews the main definitions and properties associated to conserved intervals, and Section 4 gives new fundamental results on the operations on sets of conserved intervals, together with the associated algorithms, in Section 5. In particular, we show how an algorithm, conceptually similar to the Fitch-Hartigan algorithm ([12, 15]) for character-based parsimony, can be build upon the defined set of operations. The output of this algorithm is a hypothesis regarding ancestral gene orders, in the form of a set of conserved intervals at each node of the tree. The results obtained on chloroplastic genomes, reported in Section 6, indicate that the algorithm seems effective at capturing specifically a set of likely ancestral gene orders.

2 Looking for Ancestors

We assume that gene orders are represented by signed permutations where each element corresponds to a different gene and its sign represents the gene orientation.

Definition 1 (Conserved interval [4]). *A* conserved interval *of a set of signed permutations is an interval $[a, b]$ such that a precedes b, or $-b$ precedes $-a$, in each permutation, and the set of elements, without signs, between a and b is the same in each permutation.*

Consider the following genomes P and Q represented by signed permutations on the set $\{1, 2, 3, 4, 5, 6\}$: $P = \ (1\ 2\ 3\ 4\ 5\ 6)$ and $Q = \ (1\ \ -2\ 3\ \ -5\ 4\ 6)$. The conserved intervals of P and Q are $I(\{P, Q\}) = \{[1, 3], [3, 6], [1, 6]\}$. A practical representation of conserved intervals is to choose one signed permutation of the set, box its elements, and join the extremities of conserved intervals that are not the union of smaller ones with larger boxes. For example, the conserved intervals of P and Q can be represented as:

$$I_1 = \boxed{1}\boxed{\boxed{2}}\boxed{3}\boxed{\boxed{4}\boxed{5}}\boxed{6}$$

We associate, to such a representation, the set of all signed permutations that share the same conserved intervals. Graphically, this set can be obtained by reversals and transpositions that do not "break" any box. The set $Perm(I_1)$ of signed permutations sharing the conserved intervals I_1 would thus contains 16 permutations, obtained by reversing elements 2, 4 or 5, or by transposing elements 4 and 5. (Note that the extreme points 1 and 6 are considered fixed.)

Suppose now that two other signed permutations are added in the set of genomes under study: $R = \ (1\ \ -2\ \ -3\ 5\ 4\ 6)$ and $S = \ (1\ \ -3\ 2\ 5\ \ -4\ 6)$. Their conserved intervals are represented as:

$$I_2 = \boxed{1}\boxed{\boxed{-2}\boxed{-3}}\boxed{5}\boxed{\boxed{4}}\boxed{6}$$

and the set of associated permutations contains also 16 permutations.

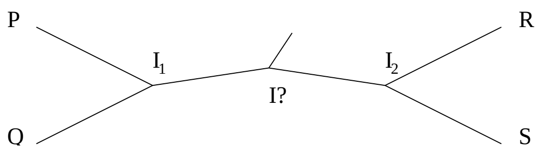
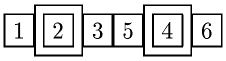


Fig. 1. A Tree Topology for the Permutations $\{P, Q, R, S\}$

If we are given the tree topology of Fig. 1, it would seem natural to label the parent of $\{P, Q\}$ with I_1, and the parent of $\{R, S\}$ with I_2. Indeed, under most reasonable rearrangement scenarios, the ancestors are respectively in $Perm(I_1)$ and $Perm(I_2)$ [4]. What should be the label I of the ancestral node? Computing the conserved intervals of the permutations $\{P, Q, R, S\}$ yields the trivial interval $[1, 6]$. However, in this example, the two sets of signed permutations associated to I_1 and I_2 have a non-empty intersection consisting of the four permutations: $(1\ 2\ 3\ 5\ 4\ 6)$, $(1\ \ -2\ 3\ 5\ 4\ 6)$, $(1\ 2\ 3\ 5\ \ -4\ 6)$, and $(1\ \ -2\ 3\ 5\ \ -4\ 6)$.

Thus, an interesting label for the ancestral node could be the set of conserved intervals of these four permutations:

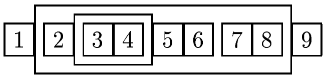
$I =$ 

Note that this set contains all conserved intervals of both sets I_1 and I_2, together with the adjacency $[3, 5]$. The distinctive characteristic of the two subgroups of the tree of Fig. 1 is the alternate ways in which the adjacency $[3, 5]$ is broken.

When the intersection of the two sets is empty, we will show, in Sections 4 and 5, that it is possible to keep a subset of each set of signed permutations, and then compute conserved intervals of the union of these sets.

3 Basic Properties of Conserved Intervals

Let G be a set of signed permutations, we will denote by $I(G)$ the set of conserved intervals of G. Sets of conserved intervals are highly structured, which was not readily apparent with the simple examples of Section 2. For example, consider the following set G of two signed permutations: $P = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$ and $Q = (1\ -6\ -5\ 3\ 4\ -2\ -8\ -7\ 9)$. Then the set $I(G)$ is represented by the following diagram, based on the permutation P.

$I(G) =$ 

A conserved interval that is not the union of smaller conserved intervals is called *irreducible*. For example, in $I(G)$, all intervals are irreducible except the interval $[2, 6]$. Irreducible intervals share at most one endpoint, as made precise by the following proposition:

Proposition 1 ([4]). *Let G be a set of signed permutations. Let $[a, b]$ and $[c, d]$ be two irreducible intervals of $I(G)$. Then $[a, b]$ and $[c, d]$ are either disjoint, nested with different endpoints, or overlapping on one element.*

Successive irreducible intervals that overlap on one element form *chains*. Chains are denoted by the successive common elements of the overlapping intervals, such as the chain $[2, 5, 6]$ in $I(G)$. It is easy to see that each conserved interval of a set of signed permutations is either irreducible, or is a chain.

Because the set of conserved intervals of a given set of signed permutations has some structural properties, a collection C of intervals is not necessarily the set of conserved intervals of a set of signed permutations. However, it is always possible to construct the smallest collection that contains C, and that is the set of conserved intervals of a set of signed permutations.

Definition 2 (Closure of a set of intervals). *Let U be a set of intervals of a signed permutation $P = (p_1, \ldots, p_n)$. The* closure *of U, denoted by U^*, is the smallest set of intervals containing U and such that, for any pair $([p_i, p_j], [p_k, p_l])$ of intervals in U^*, such that $i \leq k \leq j \leq l$, then $[p_i, p_k]$, $[p_k, p_j]$, $[p_j, p_l]$ and $[p_i, p_l]$ are in U^*, provided that they have more than one element.*

Consider the set of intervals $U = \{[1,3],[3,6],[1,5],[5,6]\}$ of the identity permutation. Its closure is given by: $U^* = \{[1,3],[1,5],[1,6],[3,5],[3,6],[5,6]\}$.

Given a set of intervals I, the maximal set of signed permutations that have all the conserved intervals of I is denoted by $Perm(I)$. Again, not all sets of signed permutations can be constructed in this way.

Definition 3 (Saturated set of permutations). *A set of signed permutations G is* saturated *if G is the set of signed permutations that have all the conserved intervals of $I(G)$, that is to say $G = Perm(I(G))$.*

For example, the set $\{(1\ \ {-2}\ \ 3\ \ 5\ \ 4\ \ 6), (1\ \ 2\ \ 3\ \ 5\ \ {-4}\ \ 6)\}$ is not saturated, because both permutations $(1\ \ 2\ \ 3\ \ 5\ \ 4\ \ 6)$, and $(1\ \ {-2}\ \ 3\ \ 5\ \ {-4}\ \ 6)$ share the same conserved intervals. These four permutations form a saturated set since they are the only ones that have the conserved intervals:

$$I = \boxed{1}\boxed{\boxed{2}}\boxed{3}\boxed{5}\boxed{\boxed{4}}\boxed{6}$$

4 Operations on Sets of Conserved Intervals

We now turn to the problem of computing the conserved intervals of unions and intersections of sets of signed permutations. The first result is the basis of a linear-time algorithm to compute the conserved intervals of the union of two sets of signed permutations.

Theorem 1 (Conserved intervals of a union [4]). *Let G_1 and G_2 be two sets of signed permutations on the set $\{1,\ldots n\}$, then $I(G_1 \cup G_2) = I(G_1) \cap I(G_2)$.*

However, there is no such simple characterization of the conserved intervals of an intersection of arbitrary sets of signed permutations. In order to have a dual property, we must assume that the sets G_1 and G_2 are saturated, but this will be the case in the algorithms we describe in Section 5.

Theorem 2 (Conserved intervals of an intersection). *Let G_1 and G_2 be two saturated sets of signed permutations on the set $\{1,\ldots n\}$. If $G_1 \cap G_2 \neq \emptyset$. Then $I(G_1 \cap G_2) = (I(G_1) \cup I(G_2))^*$.*

Note that the right hand side of the above equation is well defined, since the intersection of G_1 and G_2 is not empty, thus all intervals of $I(G_1)$ and of $I(G_2)$ can be represented using the same permutation.

Testing whether $G_1 \cap G_2$ is empty is not elementary, and is at the heart of the algorithmic complexity of constructing intersections. The next definition introduces the basic concept of *filtering* a set of signed permutations with an interval.

Definition 4 (Filtering sets of permutations). *Let $[a,b]$ be an interval of a signed permutation P, and G a saturated set of signed permutations. The* filtered *set $G_{[a,b]}$ is the subset of all signed permutations of G that have the conserved interval $[a,b]$. The set of conserved intervals of $G_{[a,b]}$ is denoted by $I(G)_{[a,b]}$.*

For example, consider the following set of conserved intervals, and the corresponding saturated set G.

$$I = \boxed{1}\,\boxed{\boxed{2}\boxed{3}\,\boxed{4}}\,\boxed{5}\,\boxed{\boxed{6}\,\boxed{7}}\,\boxed{8}.$$

Let $P = (1\ \ 3\ \ 2\ \ 4\ \ 5\ \ -7\ \ 6\ \ 8)$. Filtering G with the interval $[4,-7]$ of P yields the following set of conserved intervals:

$$I(G)_{[4,-7]} = \boxed{1}\,\boxed{\boxed{2}\boxed{3}}\,\boxed{4}\boxed{5}\boxed{\text{-}7}\,\boxed{\boxed{6}}\,\boxed{8}.$$

However, filtering G with the interval $[1,3]$ of P would yield the empty set, since no permutation of G has the conserved interval $[1,3]$.

Proposition 2. *Let G be a saturated set of signed permutations. Let $[a,b]$ and $[c,d]$ be two intervals. Then $G_{[a,b]}$ is saturated, and $(G_{[a,b]})_{[c,d]} = (G_{[c,d]})_{[a,b]}$.*

Theorem 3. *Let G_1 and G_2 be two saturated sets of signed permutations. Let $J_1, \ldots, J_k$ be the set of irreducible conserved intervals of G_1, then $G_1 \cap G_2 = \emptyset$ if and only if $(I(G_2))_{J_1,\ldots,J_k} = \emptyset$. Moreover, if $G_1 \cap G_2 \neq \emptyset$, then we have $I(G_1 \cap G_2) = (I(G_2))_{J_1,\ldots,J_k}$.*

Together with Proposition 2, this theorem yields an algorithm to compute the intersection of two saturated sets of signed permutations using successive filtering. Indeed, if there is a step in which filtering produces an empty result, then the intersection is empty.

However, even when the intersection is empty, there might still exists a non-empty subset of, say G_1, that shares conserved intervals with G_2. Such conserved intervals are likely to have been shared by a common ancestor. Some care must be taken in order to properly define these collections. Indeed sets of intervals can be *conflicting*:

Definition 5. *A set S of conserved intervals is* conflicting *with respect to a saturated set G of signed permutations if $G_S = \emptyset$ and $\forall I \in S, G_{S\setminus\{I\}} \neq \emptyset$.*

In Section 6 we will see that, when G_1 and G_2 are filtered with collections of conserved intervals in which conflicting subsets are removed, we can obtain ancestral gene orders that are extremely well-defined.

Conjecture 1. Let G_1 be a saturated set of signed permutations, and S be the set of irreducible intervals of G_1, then it is possible to identify, in polynomial time, all conflicting subsets of S with respect to a saturated set of signed permutations G_2.

5 Algorithms

We discuss now the two main algorithmic issues raised in the previous sections: filtering and ancestors labelling. We first describe how to represent a set of conserved intervals as a PQ-*tree*, then we outline a linear time filtering algorithm. Finally, we describe an ancestor labelling algorithm based on the principle of the Fitch-Hartigan parsimony algorithm.

Conserved intervals and PQ-trees. A PQ-tree is a data structure used to represent in a compact way a set of permutations [7]. Here we adapt this data structure to represent sets of conserved intervals. This idea was briefly introduced in [4].

We define a variant of PQ-*trees* as *ordered trees* with three types of nodes: n *leaves* that are labelled with signed elements of $\{1, \ldots, n\}$, and internal nodes that are either *round* or *square*. The root is always a square node, and all the children of a round (resp. square) node are square (resp. round) nodes or leaves. Moreover, among the children of a square node, the first and last are leaves and there cannot be two consecutive round nodes. It follows that the total number of nodes of a PQ-tree is linear in n. The relationship between PQ-trees and conserved intervals is as follows: a round node represents the free elements and conserved intervals that are inside a box of the box representation, and a square node represents a maximal chains of intervals. The children of a square node are either round nodes, or the endpoints of the irreducible intervals of the maximal chain it represents. See Fig. 2.

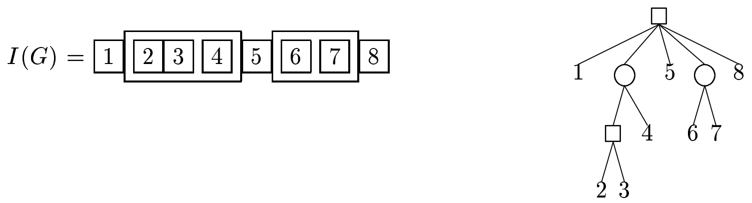


Fig. 2. Two representations of the same set of conserved intervals

Enumerating, during a depth-first traversal, the leaves of a PQ-tree representing a set I of conserved intervals, gives a permutation in $Perm(I)$. Considering PQ-trees as ordered trees implies that there are as many different PQ-trees representing I as there are permutations in $Perm(I)$, and that each of these different trees, when traversed as described above, gives a different permutation of $Perm(I)$. Indeed, performing one of the following transformations on a PQ-tree T does not change the set I of conserved intervals it represents, but implies that the new ordered tree obtained represents a different permutation of $Perm(I)$: changing the sign of a leaf incident to a round vertex, reordering the children of a round vertex, reversing the order of the children of a square vertex (except the root), and changing in the same time the signs of all leaves present among these children. It is straightforward to design simple data structures to implement PQ-trees that allow to perform transformations of a node of a PQ-tree in constant time.

Filtering a set of conserved intervals. We now outline the basic steps in the construction of the filtering algorithm[3].

Let T be a PQ-tree representing a set G of unsigned permutations on the set $\{1, \ldots n\}$ and $S \subseteq \{1, \ldots, n\}$ a given set of elements. The *reduction* of T over S yields the PQ-tree that represents the set G_S of all permutations of G in which the elements of S appear consecutively. A reduction can be computed in linear time with respect to n [7].

A conserved interval $[a, b]$ with inner elements $x_1, \ldots, x_k$ yields three sets that should appear consecutively in all permutations: $\{x_1, \ldots, x_k\}$, $\{a, x_1, \ldots, x_k\}$ and $\{x_1, \ldots, x_k, b\}$. Moreover, signed permutations on n elements can be coded by unsigned permutations on $2n$ elements by replacing $+i$ by $2i-1, 2i$, and $-i$ by $2i, 2i-1$. Thus, filtering a set G of signed permutations amounts to perform three reductions (five in the unsigned case) on the PQ-tree representing the unsigned versions of permutations in G. We have:

Proposition 3. *Let $[a, b]$ be an interval of a permutation P on n elements, and G a saturated set of signed permutations, then the set of conserved intervals of $G_{[a,b]}$ can be computed in $O(n)$ time and space.*

Ancestors labelling. We now describe an algorithm for inferring putative ancestral genes orders for a phylogenetic tree with a given topology and with gene orders at the leaves. The algorithm is similar in spirit to the Fitch-Hartigan algorithm ([12], [15]) for character-based parsimony, and consists of two labelling phases: a bottom-up labelling and a top-down refinement of this labelling.

Bottom-up labelling. In a first phase, during a bottom-up traversal of the tree, each ancestral node is labelled with a set of conserved intervals and the associated saturated set of signed permutations. Let x be a node with children y and z, and assume that y and z are already labelled by saturated sets of signed permutations G_y and G_z, with sets of conserved intervals I_y and I_z. Intuitively, we choose the label I_x that has as many intervals in common with I_y and I_z as possible. If $G_y \cap G_z \neq \emptyset$, then we set $I_x = I(G_y \cap G_z)$. If $G_y \cap G_z = \emptyset$, then I_y and I_z contain some conflicting intervals that need to be removed. We first identify S_y the subset of I_y that contains intervals that do not belong to conflicting subsets with respect to G_z. We obtain S_z similarly, with respect to G_y. Finally, we set $I_x = I((G_y)_{S_z} \cup (G_z)_{S_y})$ and $G_x = Perm(I_x)$. The algorithm proceeds up the tree until a label for the root is obtained.

Top-down refinement. While the root of the tree was assigned a label I_{root} based on all the leaves of the tree, this is the not the case for the other internal nodes, which were so far inferred based only on the leaves of the subtree of which they are the root. To let the information about all leaves be used to establish ancestral genomes, we proceed to a second phase, again similar to the second phase of the Fitch-Hartigan algorithm, where the conserved interval I_x of node x are used to refine the conserved intervals of the children of x. For any child y of node x, we

[3] The full technical details of the implementation of this algorithm will appear in [2].

first compute S_{xy}, the subset of I_x that contains intervals that do not belong to conflicting subsets with respect to G_y. We then refine I_y as $I_y = (I_y \cup S_{xy})^*$, and obtain $G_y = Perm(I_y)$.

By Theorem 2 and assuming that Conjecture 1 holds, the running time of the whole labelling procedure is polynomial in the number of genes and the number of leaves of the tree.

6 Chloroplast genomes

To assess the specificity of the ancestral gene order reconstruction method described above, we tested our algorithm on a subset of gene segments of the chloroplast genomes of 13 species of plants previously studied by Cosner et al.[10].

Based on a phylogenetic tree previously reported for these species, the two phases of the ancestral gene order reconstruction were performed, and the inferred sets of conserved intervals are illustrated in Fig. 4. For example, in the first phase of the algorithm, when building the set of conserved intervals for the ancestor of Legousia and Triodanus: since both sets contain single permutations, the label of the ancestor is $I(G_{Leg} \cup G_{Tri})$. This yields the following representation of the conserved intervals:

$$A_{Leg,Tri} = \boxed{0}\ \boxed{\boxed{1\ 2}\ \boxed{10\ 7\ 11\ 12}}\ \boxed{13\ \ 14\ 3\ \text{-}9\ \text{-}8\ \text{-}6\ \text{-}5\ \text{-}4\ 15}.$$

Then, using this reconstruction to build the ancestor of Asyneuma, we note that intervals [0,1] and [2,13] of Asyneuma are conflicting with respect to $A_{Leg,Tri}$. The resulting set of compatible conserved elements is represented by:

$$A_{Leg,Tri,Asy} = \boxed{0}\ \boxed{\boxed{1\ 2}\ \boxed{10\ 7\ 11\ 12}\ \boxed{13\ 14\ 3}}\ \boxed{\text{-}9\ \text{-}8\ \text{-}6\ \text{-}5\ \text{-}4\ 15}.$$

The process continues up the tree until the ancestral gene order at the root of the tree is obtained. Fig. 3 shows the resulting set of conserved intervals. The second phase of the algorithm then starts and the information is propagated down the tree, starting from the root and adding conserved intervals to the children as often as possible. The resulting sets of conserved elements, shown in Fig. 4, is often much more refined than those obtained during the first phase of the algorithm. For example, the two ancestors $A_{Leg,Tri}$ and $A_{Leg,Tri,Asy}$ are now pinpointed to single possible permutations, separated by one reversal.

A closer inspection of the ancestral gene orders reconstructed reveals that, although the criterion used for inferring them was not based on a notion of parsimony of genome rearrangements, the distance, in terms of number of rearrangements, between neighboring sets of ancestral conserved intervals is usually very small, and often zero. We observe that most rearrangements that can be deduced from the reconstructed ancestors are reversals, but that a few transpositions and inverted transpositions also occur, for example between $A_{Leg,Tri}$ and Legousia or between Platycodon and its ancestor.

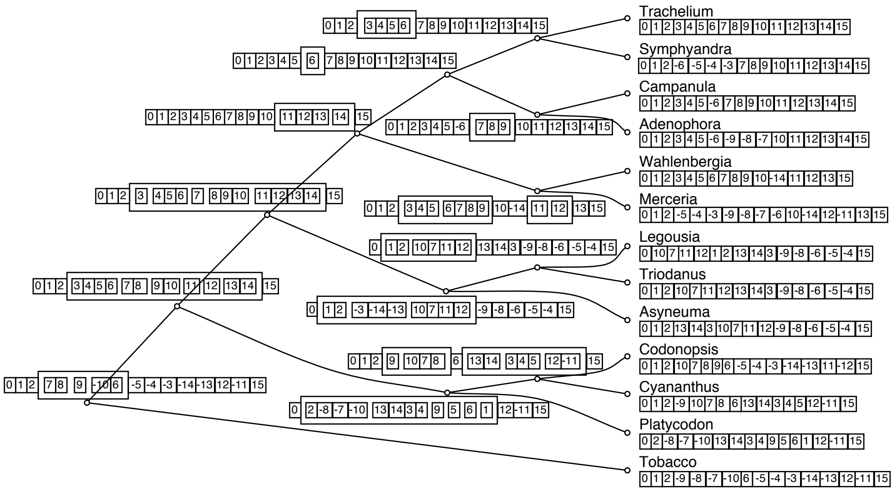


Fig. 3. Reconstructed conserved intervals for internal nodes of the campanulaceae phylogeny, after bottom-up labelling.

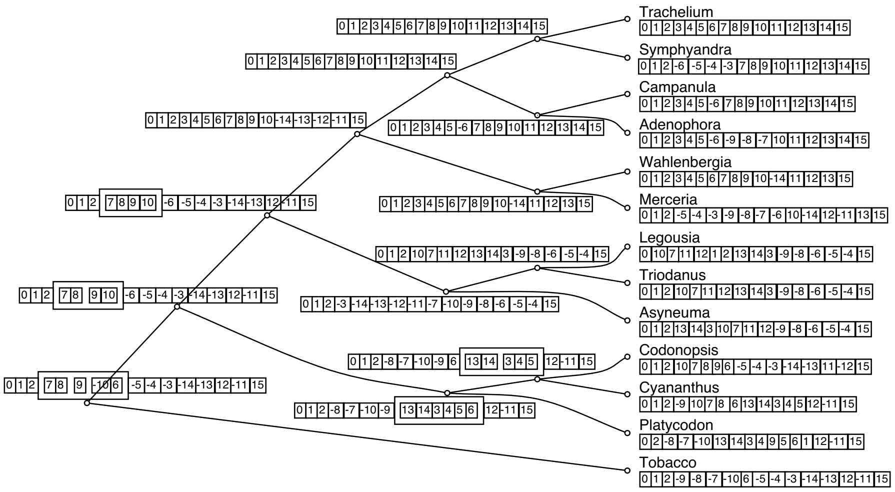


Fig. 4. Reconstructed conserved intervals for internal nodes of the campanulaceae phylogeny, after top-down refinement.

7 Conclusion

This paper presented operations on sets of conserved intervals, as well as associated techniques applied to the reconstruction of ancestral gene orders. The results obtained on a classical data set based on chloroplast genomes are very encouraging.

The next step is to apply our algorithms to the inference of complete ancestral mitochondrial and chloroplast genomes, and eventually to whole nuclear genomes. This would yield a better understanding of the rearrangement processes at work in these genomes. It may also highlight some highly conserved intervals that may correspond to sets of genes with strong positional ties, such as operons in bacteria.

However, the method presented here has still to be validated by using on simulated data. Given phylogenetic trees with known ancestral gene orders, we will test our algorithms on these trees and compare the results to the original ancestral gene orders.

References

1. D.A. Bader, B.M.E. Moret, and M. Yan. A linear-time algorithm for computing inversion distances between signed permutations with an experimental study. *J. Comput. Biol.*, 8(5):483–491, 2001.
2. A. Bergeron, M. Blanchette, A. Chateau, and C. Chauve. Implementation of operations on sets of conserved intervals. Technical report, Computer Science Department, UQAM, To appear.
3. A. Bergeron, J. Mixtaci, and J. Stoye. The reversal distance problem. In O. Gascuel, editor, *Mathematics of phylogeny and evolution.* Oxford University Press, To appear in 2004.
4. A. Bergeron and J. Stoye. On the similarity of sets of permutations and its applications to genome comparison. In *9th Annual International Conference on Computing and Combinatorics (COCOON 2003)*, volume 2697 of *Lecture Notes in Comput. Sci.*, pages 68–79. Springer, 2003.
5. M. Blanchette, T. Kunisawa, and D. Sankoff. Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.*, 19(2):193–203, 1999.
6. M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Gene*, 172(1):GC11–17, 2001.
7. K.S. Booth and G.S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using *PQ*-tree algorithms. *J. Comput. System Sci.*, 13(3):335–379, 1976.
8. G. Bourque and P.A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res.*, 12(1):26–36, 2002.
9. A. Caprara. Formulations and complexity of multiple sorting by reversals. In *3rd Annual International Conference on Research in Computational Molecular Biology (RECOMB 1999)*, pages 84–93. ACM Press, 1999.
10. M.E. Cosner, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, L.S. Wang, T. Warnow, and S.K. Wyman. An empirical comparison of phylogenetic methods on chloroplast gene order data in campanulaceae. In D. Sankoff and J. Nadeau, editors,

Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families (DCAF 2000), pages 99–212. Kluwer Academic Publishers, 2000.
11. J. Felsenstein. *Inferring phylogenies.* Sinauer Associates, 2003.
12. W. Fitch. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20:406–416, 1971.
13. S. Hannenhalli and P.A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *36th Annual Symposium on Foundations of Computer Science (FOCS 1995)*, pages 581–592. IEEE Comput. Soc. Press, 1995.
14. S. Hannenhalli and P.A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J. ACM*, 46(1):1–27, 1999.
15. J. A. Hartigan. Minimum mutation fits to a given tree. *Biometrics*, 29:53–65, 1973.
16. B. Larget, D.L. Simon, and J.B. Kadane. Bayesian phylogenetic inference from animal mitochondrial genome arrangements. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 64(4):681–693, 2002.
17. B.M.E. Moret, A.C. Siepel, J. Tang, and T. Liu. Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. In *2nd International Workshop on Algorithms in Bioinformatics (WABI 2002)*, volume 2452 of *Lecture Notes in Comput. Sci.*, pages 521–536. Springer, 2001.
18. B.M.E. Moret, J. Tang, and T. Warnow. Reconstructing phylogenies from gene-content and gene-order data. In O. Gascuel, editor, *Mathematics of phylogeny and evolution.* Oxford University Press, To appear in 2004.
19. B.M.E. Moret, S. Wyman, D.A. Bader, T. Warnow, and M. Yan. A new implementation and detailed study of breakpoint analysis. In *6th Pacific Symposium on Biocomputing (PSB 2001)*, pages 583–594, 2001.
20. D. Sankoff. Rearrangements and chromosomal evolution. *Curr. Opin. Genet. Dev.*, 13(6):583–587, 2003.
21. D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *J. Comput. Biol.*, 5(3):555–570, 1998.
22. D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B.F. Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. *Proc. Natl. Acad. Sci. U.S.A.*, 89(14):6575–6579, 1992.
23. J. Tang and B.M.E. Moret. Scaling up accurate phylogenetic reconstruction from gene-order data. *Bioinformatics*, 19(Suppl. 1):i305–312, 2003.