

Common Intervals and Symmetric Difference in a Model-Free Phylogenomics, with an Application to Streptophyte Evolution

Zaky Adam¹, Monique Turmel², Claude Lemieux², and David Sankoff¹

¹ School of Information Technology and Engineering
and Department of Mathematics and Statistics,
University of Ottawa, Ottawa, Canada, K1N 6N5,
{zadam008, sankoff}@uottawa.ca

² Département de biochimie et de microbiologie,
Université Laval, Québec, Canada, G1K 7P4,
{monique.turmel, claude.lemieux}@rsvs.ulaval.ca

Abstract. The common intervals of two permutations on n elements are the subsets of terms contiguous in both permutations. They constitute the most basic representation of conserved local order. We use d , the size of the symmetric difference (the complement of the common intervals) of the two subsets of $2^{\{1, \dots, n\}}$ thus determined by two permutations, as an evolutionary distance between the gene orders represented by the permutations. We consider the Steiner Tree problem in the space $(2^{\{1, \dots, n\}}, d)$ as the basis for constructing phylogenetic trees, including ancestral gene orders. We extend this to genomes with unequal gene content and to genomes containing gene families. Applied to streptophyte phylogeny, our method does not support the positioning of the complex algae *Charales* as a sister group to the land plants.

1 Introduction

Phylogenetics based on gene order has used two types of objective functions for optimizing the inferred ancestral nodes. One is based on breakpoints, or non-conserved adjacencies in the gene orders of two genomes, dating from 1997 [15] and related to the quantitative pre-genomics literature on conserved segments [12]. The second is based on the number of inversions or other genome rearrangements intervening between two genomes [18,7,19,6].

The scope of a genome rearrangement operation may be unrestricted across the genome; a breakpoint is a very local structure. To emphasize local similarities spanning regions larger than a single breakpoint within a rearrangement analysis, Bergeron and colleagues have integrated more general notions of conserved intervals and common intervals with rearrangements [3,4]. Further, they used these integrated concepts in phylogeny, including the reconstruction of ancestral gene orders [1,2].

In this paper we put even more importance on common intervals, basing a phylogenetic reconstruction method solely on them. This analysis is model-free,

in the sense that it dispenses completely with assumptions or considerations, probabilistic or combinatorial, about specific processes involved in rearranging genomes. The objective function we will optimize is simply the sum over the tree branches of the symmetric difference between the two sets of intervals associated with the genomes at the two ends of the branch. The motivation is simply that the more related two genomes are, the more common intervals they will have, while as evolutionary distance increases, more and more intervals from each will be lacking from the other.

Optimizing a tree on the space of subsets of the power set of $\{1, \dots, n\}$ using dynamic programming [8,9,17] is easy; the set of intervals representing a genome, however, is constrained to be compatible with a permutation. We do not know the complexity of optimizing the ancestral nodes of a tree under this constraint, but conjecture that it is hard. Thus we model our optimization heuristic after the unconstrained case, and add the constraint in the traceback of the algorithm, where a greedy procedure is used to test the addition of possibly conflicting intervals to the subset of intervals representing a genome, represented by a PQ-tree.

Our methods carry over directly to the case where some or all genomes do not contain the full set of genes. All we require is a preliminary assignment of genes to ancestral nodes, rapidly achieved by dynamic programming, and an adjustment of a test for identity of two intervals to be restricted to the reduced intervals containing only the genes in common in the two genomes. The same sort of extension of the model works when duplicate genes and gene families are allowed, but only under certain conditions; the general case will require further work.

We apply our method to the controversial questions of streptophyte phylogeny and recover results comparable to previous work on both gene order phylogeny and DNA sequence-based phylogeny.

2 Notation and Definitions

For a permutation Π on $(1, \dots, n)$, we write $\Pi = (\pi(1), \dots, \pi(n))$. Let $\mathcal{S} = \{\Pi_1, \dots, \Pi_N\}$ be a set of such permutations. For each Π , the interval set determined by $\pi(h)$ and $\pi(k)$, for $1 \leq h < k \leq n$, is $\{\pi(h), \pi(h+1), \dots, \pi(k)\}$. For each $J = 1, \dots, N$, let $\mathcal{I}_J \subset 2^{\{1, \dots, n\}}$ be the $\binom{n}{2}$ interval sets of $2^{\{1, \dots, n\}}$ determined by Π_J . We define the projection $B(\Pi_J) = \mathcal{I}_J$. The set of common intervals of Π_J and Π_K is $B(\Pi_J) \cap B(\Pi_K) = \mathcal{I}_J \cap \mathcal{I}_K$, the intersection of the $\binom{n}{2}$ interval sets defined by Π_J and those defined by Π_K .

Let $X \subseteq \mathcal{I}_J$ and $Y \subseteq \mathcal{I}_K$. We say X is compatible with $B(\Pi_J)$ and Y is compatible with $B(\Pi_K)$ and define the metric

$$\begin{aligned} d(X, Y) &= |X \cup Y \setminus X \cap Y| \\ &= |X| + |Y| - 2|X \cap Y|. \end{aligned} \tag{1}$$

In particular,

$$\frac{1}{2}d(\mathcal{I}_J, \mathcal{I}_K) = \binom{n}{2} - |\mathcal{I}_J \cap \mathcal{I}_K|. \tag{2}$$

Note that not all subsets of $2^{\{1, \dots, n\}}$ are permutation-compatible, i.e., are subsets of $B(\Pi)$ for some permutation Π . E.g., $\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$ is not permutation-compatible. For any permutation-compatible X , we define $B^{-1}(X)$ to be the set of permutations Π for which $X \subseteq B(\Pi)$.

3 The Steiner Tree Problem for Common Intervals

The Steiner tree problem with input \mathcal{S} is to find a tree graph $T = (V, E)$, where the vertices in V are permutation-compatible subsets of $2^{\{1, \dots, n\}}$ and $\{B(\Pi)\}_{\Pi \in \mathcal{S}} \subseteq V$ such that the tree length

$$L(T) = \sum_{XY \in E} d(X, Y) \quad (3)$$

is minimal.

In a Steiner tree, we can distinguish two types of vertex in V , terminal vertices, i.e., of degree 1, which are all projections of permutations in \mathcal{S} , and non-terminal, or “ancestral”, vertices, some or all of which, the “unknown vertices” may be projections of permutations not in \mathcal{S} or other permutation-compatible subsets of $2^{\{1, \dots, n\}}$. We require that unknown vertices have degree 3 or more, a condition which, by the metric property, does not affect the minimal value of L in (3).

The search for the Steiner tree is often divided into two problems, the inner, or “small”, problem, and the outer, or “big”, problem. The big problem is essentially a search through the set of all trees satisfying the above description. For each tree examined during this search, the small problem is to optimally identify each of the unknown vertices in V as the projection of some permutation on $(1, \dots, n)$ or some other permutation-compatible subset of $2^{\{1, \dots, n\}}$.

4 The General Dynamic Programming Solution for the Small Problem

A general method for the small problem, i.e., for optimizing the position of the ancestral nodes of a tree in a metric space was given in [17].

Condition 1. It suffices to consider trees where there is a path between any two unknown vertices not passing through a vertex in \mathcal{S} ; otherwise the problem decomposes into subproblems an obvious way.

The solution requires choosing, arbitrarily, one of the unknown vertices r as the “root”, and directing all edges in E away from the root. We write $v \rightarrow u$ for an edge directed from ancestor v to “daughter” u . For any given position of the ancestral vertices, let

$$l(v) = \sum_{v \rightarrow u} l(u) + d(u, v), \quad (4)$$

with initial condition $l(v) = 0$ if v is a terminal vertex. Note that the tree structure and Condition 1 ensure that recurrence (4) determines $l(v)$ for all non-terminal vertices. Then for any tree T with specified positions of the ancestral vertices we may rewrite (3) as

$$L(T) = l(r). \quad (5)$$

For a Steiner tree, for a given position of v , we obtain from (4)

$$l(v) = \sum_{v \rightarrow u} \min_u [l(u) + d(u, v)]. \quad (6)$$

If all the minimizing u are stored at each application of (6), then a Steiner tree T may be recovered by tracing back the recurrence from the root to the terminal nodes.

Depending on the metric space, the minimizing u in (6) may be more or less difficult to calculate. Such is the case that interests us here, where the vertices are projections of permutations, and the search for optimizing u is not straightforward. However, we can easily solve a closely related problem, which provides a lower bound on $L(T)$ and suggests how to solve the problem on permutations.

5 Embedding the Small Problem for Permutations in a Larger Space

For a given \mathcal{S} consider the Steiner tree problem with input $\{B(\Pi_1), \dots, B(\Pi_N)\}$ in the metric space (M_n, d) , where $M_n = 2^{2^{\{1, \dots, n\}}}$, the set of all subsets of the power set of $\{1, \dots, n\}$, and d is as before.

The set M_n is larger than \mathcal{P}_n , the set of all permutation-compatible subsets of $2^{\{1, \dots, n\}}$, in that there are elements of $X \in M_n$ which are not of the form $X \subseteq B(\Pi)$ for any permutation Π .

Example 1. Let $X = \{\{1, 2\}, \{1, 3\}, \{1, 4\}\}$. Then there is no permutation Π for which $X \subseteq B(\Pi)$ for any permutation Π .

Then the solution to the Steiner tree problem in (M_n, d) is a lower bound to the solution of the problem in (\mathcal{P}_n, d) .

The Steiner tree problem in (\mathcal{P}_n, d) is harder than in (M_n, d) because the intervals sets of a permutation, or sets compatible with a permutation, are highly constrained, whereas in (M_n, d) it suffices to treat each subset of $\{1, \dots, n\}$ separately. To see this, note that the symmetric difference between two points $X, Y \in M_n$ may be written

$$d(X, Y) = \sum_{\sigma \in \{1, \dots, n\}} |\chi_\sigma(X) - \chi_\sigma(Y)|, \quad (7)$$

where χ_σ is the indicator function of σ . Then, writing $X(v)$ for the element of M_n associated with vertex v of a tree, the length of any tree T satisfies

$$L(T) = \sum_{uv \in E} d(X(u), X(v)) \quad (8)$$

$$= \sum_{uv \in E} \sum_{\sigma \subset \{1, \dots, n\}} |\chi_\sigma(X(u)) - \chi_\sigma(X(v))| \quad (9)$$

$$= \sum_{\sigma \subset \{1, \dots, n\}} \sum_{uv \in E} |\chi_\sigma(X(u)) - \chi_\sigma(X(v))| \quad (10)$$

so that the minimization of $L(T)$ may be done component-wise, i.e., separately for each $\sigma \subset \{1, \dots, n\}$. In Section 6 then, we will discuss only whether or not each ancestral vertex contains σ or not, which can be phrased as the tree optimization problem for a single zero-one character.

6 The Small Problem for a Single Zero-One Variable

Consider that the data at each terminal vertex consist of either a zero or a one, representing the presence or absence of a set σ . Dynamic programming for ancestral node optimization requires two passes. In the forward pass, from the terminal nodes towards the root, the value of the variable (the presence or absence of σ) may be established definitely at some ancestral vertices, while at other vertices it is left unresolved until the second, “traceback” pass, when any multiple solutions are also identified.

Suppose ancestral vertex v has p daughter vertices u_1, \dots, u_p , where $p \geq 2$. (If $v \neq r$, then v has degree $p + 1$, if $v = r$, then v has degree $p \geq 3$.) In practice, it usually suffices to allow only $p = 2$ for ancestral vertices and $p = 3$ for the root (binary trees), and indeed our algorithm is programmed for this case. Nevertheless, in this section we will give the general solution, which may be required in some contexts, and is of mathematical interest in any case. In the next section we will discuss the simplification to binary trees.

Suppose for each daughter u_i , we have already decided whether $\sigma \in u_i$ definitely, possibly, or definitely not. Let

$$q(\sigma) = \#\{i | \sigma \in u_i \text{ definitely}\} \quad (11)$$

$$Q(\sigma) = \#\{i | \sigma \in u_i \text{ definitely or possibly}\}. \quad (12)$$

If $v \neq r$ and $q(\sigma) \geq \frac{p}{2} + 1$, then $\sigma \in v$ definitely. This is true because then

$$\sum_{v \rightarrow u} |\chi_\sigma(u) - \chi_\sigma(v)| \leq \frac{p}{2} - 1 \quad (13)$$

whereas if σ were not in v ,

$$\sum_{v \rightarrow u} |\chi_\sigma(u) - \chi_\sigma(v)| \geq \frac{p}{2} + 1 \quad (14)$$

which is clearly not optimal, no matter whether σ is in the ancestor of v or not.

On the other hand, if $v \neq r$ and $Q(\sigma) \leq \frac{p}{2} - 1$, then definitely $\sigma \notin v$. This is true because then

$$\sum_{v \rightarrow u} |\chi_\sigma(u) - \chi_\sigma(v)| \leq \frac{p}{2} - 1 \tag{15}$$

whereas if σ were in v ,

$$\sum_{v \rightarrow u} |\chi_\sigma(u) - \chi_\sigma(v)| \geq \frac{p}{2} + 1 \tag{16}$$

which is clearly not optimal, no matter whether σ is in the ancestor of v or not.

This leaves the cases where both

$$q(\sigma) < \frac{p}{2} + 1 \tag{17}$$

$$Q(\sigma) > \frac{p}{2} - 1, \tag{18}$$

where we say $\sigma \in v$ possibly.

As for r , if $q(\sigma) > \frac{p}{2}$, then $\sigma \in r$ definitely; if $Q(\sigma) < \frac{p}{2}$, then $\sigma \notin r$ definitely; otherwise $\sigma \in r$ possibly.

While applying the traceback of the recurrence, we reassign the “possible” memberships in ancestral vertices either to definite memberships or to definite exclusions. For ancestral vertex v , whether or not $\sigma \in t$, the ancestor of v , has no bearing if $\sigma \in v$ definitely or $\sigma \notin v$ definitely. Otherwise, if $\sigma \in t$ and $1 + q(\sigma) > \frac{p+1}{2}$, then we reassign $\sigma \in v$ definitely. If $\sigma \notin t$ and $Q(\sigma) < \frac{p+1}{2}$, then we reassign $\sigma \notin v$ definitely.

In the remaining cases if $\sigma \in t$ and $1 + q(\sigma) \leq \frac{p+1}{2} \leq 1 + Q(\sigma)$, or if $\sigma \notin t$ and $Q(\sigma) \geq \frac{p+1}{2} \geq q(\sigma)$, then we can assign or exclude σ from v , without affecting $L(T)$. Note that if $\sigma \in r$ possibly, we are also free to assign it to r or not at the beginning of the traceback.

Note that while the dynamic programming can find a solution in time linear in N , the number of different solutions may be exponential in N . Considering all possible sets σ , the number of solutions is also exponential in n .

7 Binary Trees

The case of binary branching trees, where $p + 1 \equiv 3$ except at the root where $p = 3$, is somewhat simpler in the traceback as there is at most one free choice of assignment, and that is at r . For any other ancestral vertex v , membership or not of σ in t , the ancestor of v , always determines its membership, or not, in v .

Moreover, for the big phylogeny problem, it suffices to search for the optimal binary tree. If the optimal tree is not binary, this will still show up as a binary tree with one or more edges of length $d = 0$.

Nevertheless, even for the small problem, because of the possible choice at r , when we consider all σ , the number of solutions may be exponential in n .

8 Handling Incompatible Subsets in \mathcal{P}_n with PQ-Trees

In the case of binary trees, after the forward pass of the dynamic programming, those σ for which $q(\sigma) = 2$ at any ancestral vertex v must all be in $B(\Pi)$ for some permutation Π , namely any terminal vertex permutation Π for which v is an ancestor. In the case of r , those σ for which $q = 3$ must be in the interval sets of all the terminal vertex permutations in \mathcal{S} . In this special case, the solution in (M_n, d) is also a solution for (\mathcal{P}_n, d) .

In general, however, this is not the case. In the following example the Steiner tree in M_n is shorter than that in \mathcal{P}_n .

Example 2. Let $\mathcal{S} = \{(4, 1, 2, 3), (2, 1, 3, 4), (3, 1, 4, 2)\}$, and suppose T has a single ancestor r .

Then in M_n , we calculate

$$\begin{aligned} q(\{1, 2\}) &= q(\{1, 3\}) = q(\{1, 4\}) = 2, \\ q(\{1, 2, 3\}) &= q(\{1, 3, 4\}) = q(\{1, 4, 2\}) = 2, \\ q(\{2, 3, 4\}) &= 0, \\ q(\{2, 3\}) &= q(\{3, 4\}) = q(\{4, 2\}) = 1, \end{aligned}$$

so that the only proper subsets in r are $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 3, 4\}, \{1, 4, 2\}$. Each of the terminal vertices is missing two of these subsets but contains one that is not in r , so that $\sum d = 3 \times (2 + 1) = 9$.

In \mathcal{P}_n , there are multiple solutions to optimizing r , including the permutations $(4, 1, 2, 3), (2, 1, 3, 4)$ or $(3, 1, 4, 2)$ themselves. Each one of the latter shares only two interval sets with each other, so that $\sum d = 12$. Thus the difficulty with working in \mathcal{P}_n is the requirement that the set of subsets at an ancestral vertex has to correspond to a permutation, or at least be compatible with some permutation, while this restriction does not apply in M_n .

There is a data structure particularly well-suited for representing a set of subsets compatible with a permutation, namely the PQ-tree [5]. A PQ-tree is just a rooted tree with terminal vertices $1, \dots, n$, where the daughter vertices of some special non-terminal vertices may be “ordered”, though the left-right or right-left direction of the order is not specified. For an ordered vertex x , the blocks of terms spanned by the various daughters of x must be in the same order for any permutation compatible with the tree, while there is no such constraint on the vertices which are not ordered.

Given a PQ-tree and a new subset X , it is possible to rapidly check whether X is compatible with the other subsets previously used to build the PQ-tree and to update the PQ-tree to include the additional ordering information, if any, in X . PQ-trees have already been utilized in gene order phylogenetics [2,14].

To ensure optimality in our procedure, we would have to interpret u and v in recurrence (6) as PQ-trees. The difficulty would be to carry out the minimization over all possible u_1, \dots, u_p , i.e., to find a constraint limiting the set of possible PQ-trees for u that could be in a solution if a given PQ-tree for v is. This is a problem even for binary trees; the search space of all possible pairs of PQ-trees cannot be reduced to a very limited set as we did in Equation (10) and Section 6. The actual calculation of the symmetric difference induced by two PQ-trees

is not time-consuming. One approach to this problem might be found in the concept of PQR-trees [11,21], which could include a limited number of intervals incompatible with one of the two descendants of v , but we have not explored this.

Instead, we proceed heuristically, constructing a PQ-tree at each vertex only during the traceback. This is motivated by the observation in our test data, that in the traceback it is rare for the following configuration to occur:

- t is the ancestor of v , and v is the ancestor of u_1 and u_2 .
- σ_1 and σ_2 are both definitely in the PQ-tree at t , after the traceback step at t .
- σ_i is in u_j only for $i = j$ but not for $i \neq j$, before the traceback.
- either one of σ_1 and σ_2 can be added to PQ-tree defined by the pre-rollback definites at t , but not both.

If this never happens, then we can be sure our result is optimal. If it does happen, we assign as many such σ to the definites of v as possible, using a greedy approach with larger intervals tried before smaller intervals.

9 Unequal Gene Complements

Doing gene order phylogeny on realistic sets of genomes which contain different sets of genes is recognized as a substantially more difficult than the case with identical gene sets in all species [16,20].

There is a natural solution within our framework. There are two steps to this solution.

First, the presence or absence of each gene at each vertex is determined by dynamic programming to minimize the number of gene deletions or insertions across all edges of the tree. This is done gene-by-gene, analogous to the set-by-set procedure in Section 5.

Second, in our main algorithm, when the definites for a vertex v are being decided in the recurrence step, any interval σ in one of its descendants u_1 containing a gene g absent from the other genome u_2 is assigned to be a definite of v if $\sigma \setminus \{g\}$ is present in u_2 . This includes the case where $\sigma = \{g, h\}$ and gene h has previously been decided to be present in u_2 .

In building the PQ-tree for u_2 during the traceback, if σ is definite for v , and $\sigma \setminus \{g\}$ is a possible for u_2 , then it is a candidate for inclusion in the PQ-tree. In calculating the symmetric difference between the sets of intervals from two genomes, we first delete from consideration any gene that is not present in both genomes, and remove any null sets or duplicate sets.

It can be shown that this generalizes the minimization of the sum of symmetric differences across the tree, given the correctness of our dynamic programming solution for gene presence or absence at ancestral vertices. In particular, in the case of identical gene complements, it reduces to our original method.

10 Duplicate Genes

The introduction of gene duplicates and gene families into genome comparisons causes even more difficulty than unequal gene complements. This either calls for changing the analytical framework from comparing permutations to comparing strings, or integrating gene-tree/species-tree methodology into our procedures, or both. The method proposed in this paper, however, formally applies directly to data containing occasional duplicate genes. The construction and comparison of sets of intervals is not complicated by duplicates, except that some convention must be adopted for intervals containing two or more copies of the same gene, i.e., only include one copy per interval or all copies, requiring the same number of copies for identity of two intervals. We did not encounter this problem with our test data, so we have not yet explored it further.

11 Application to Chloroplast Genomes

New sequences of the chloroplast genome allow the exploration of the evolution of the streptophytes, a phylum containing several classes of algae but also the familiar multicellular land plants. Our data includes gene orders from *Mesostigma viride* [10], *Chlorokybus atmophyticus* [unpublished data], *Staurastrum punctulatum* [23], *Zygnema circumcarinatum* [23], *Chaetosphaeridium globosum* [22], *Chara vulgaris* [24], as well as the land plant *Marchantia polymorpha* [13], with 120-140 genes per genome. The first six of these represent five of the six major charophycean (i.e., other than land plants) lineages.

We use the complete gene order of these genomes, including duplicate genes and genes not present in some organisms. There are 148 distinct genes, 35 of which were absent from one or more of the genomes. Five of the genomes had six or eight duplicated genes, largely the same ones in each, while the remaining two genomes had one or zero duplications, respectively.

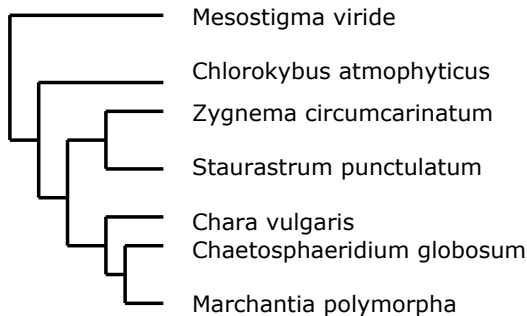


Fig. 1. Best tree for six algae and the land plants. Rooting and branch lengths arbitrary.

There has been some controversy among phycologists about the origin of land plants, in particular whether they represent a sister grouping to the Charales, represented here by *Chara vulgaris* or a sister grouping to the Coleochaetales, represented here by *Chaetosphaeridium globosum*. The best tree we obtained is depicted in Figure 1. We note the correct grouping together of the two Zygnematalean algae *Staurastrum* and *Zygnema*. The tree also correctly depicts the early branching of *Mesostigma* and *Chlorokybus*, though it does not bear on the controversy of whether *Mesostigma* is actually a streptophyte, or whether its origin predates that of the streptophytes in algal evolution. Of particular interest is the grouping of *Marchantia* with *Chaetosphaeridium* instead of with *Chara*. Previous analysis of the same genomes, at both the sequence and gene-order levels, suggested that *Chara* branches early and that the Zygnematale-Coleochaetale lineages group with the land plants [24]. The early branching of the Zygnematales was seen to be a much less parsimonious solution. Nevertheless, the results in Figure 1 represent an additional line of evidence in the still unsettled problem of streptophyte phylogeny.

12 Implementation

Our current implementation of the method includes an exhaustive search of binary trees only for the “big” problem, plus heuristics when N is larger than 9 or 10. The algorithm for the “small” problem is also the binary tree version. We have tested it for values of n in the range of 100-200. The PQ-tree construction notifies when a conflict is detected and resolved, and the pre-calculation of gene content of ancestral nodes is also implemented. The ability to handle duplicate genes is inherent in the algorithm and requires no special consideration. The experimental version of our program will be made available on our lab website.

13 Conclusions

We have proposed and implemented a new method based on common intervals of two or more genomes for constructing a phylogenetic tree. This has the advantage (or disadvantage!) of being independent of any particular model of genome rearrangement or rearrangement weighting. A maximum of emphasis is laid on the commonality of gene order at the local level, the objective function for the tree being merely the sum, over all the tree branches, of the symmetric difference between the two sets of intervals associated with the genomes at the two ends of the branch.

We have only begun to explore the algorithmic possibilities in this approach. The use of PQR-trees during the recurrence part of the algorithm might allow for a global and efficient optimum in the general case. Failing that, more sophisticated heuristics are certainly available for the construction of PQ-trees at ancestral vertices during the traceback.

One issue we have not examined is the interpretation of branch length. The number of intervals at terminal vertex is $O(n^2)$, but in our test data the number

of intervals at ancestral vertices is typically $O(n)$, so that terminal branches would appear unduly distant from the cluster of ancestral vertices, were the untransformed symmetric distance considered meaningful as a clocklike measure of evolution.

Note that this paper deals only with unsigned genomic data. There is no particularly natural way to extend it to signed genomes, except of course to simply drop the sign. Our approach, however, does seem particularly well-suited to situations with gene families, and even to string, instead of permutation, representations of genomes.

Acknowledgements

We would like to thank Laxmi Parida and Annie Chateau for useful discussions on PQ-trees. Research supported in part by a grant to MT,CL and DS from the Natural Sciences and Engineering Research Council of Canada (NSERC) as well as individual grants from this agency. DS holds the Canada Research Chair in Mathematical Genomics and is a Fellow of the Evolutionary Biology Program of the Canadian Institute for Advanced Research.

References

1. Bérard, S., Bergeron, A. and Chauve, C. 2005. Conservation of combinatorial structures in evolution scenarios. Proceedings of the Second RECOMB Satellite Workshop on Comparative Genomics (RECOMB CG 2004), J. Lagergren, ed. Lecture Notes in Computer Science 3388, Springer, 1–14.
2. Bergeron, A., Blanchette, M., Chateau, A. and Chauve, C. 2004. Reconstructing ancestral gene orders using conserved intervals. Proceedings of the First Workshop on Algorithms in Bioinformatics (WABI 2004), Lecture Notes in Computer Science 3240, Springer, 14–25.
3. Bergeron, A., Heber, S. and Stoye, J. 2002. Common intervals and sorting by reversal: a marriage of necessity. *Bioinformatics* 18: S54–S63.
4. Bergeron, A. and Stoye, J. 2003. On the similarity of sets of permutations and its applications to genome comparison. Proceedings of the Ninth International Computing and Combinatorics (COCOON 2003), Lecture Notes in Computer Science 2697, Springer, 69–79.
5. Booth K. and Luekar G. 1976. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences* 13:335–379.
6. Bourque, G. and Pevzner, P.A. 2002. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Research* 12:26–36.
7. Caprara, A. 2001. On the practical solution of the reversal median problem. Proceedings of the First Workshop on Algorithms in Bioinformatics (WABI 2001), Lecture Notes in Computer Science, Springer, 238–251.
8. Fitch, W. 1971. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology* 20: 406–416.
9. Hartigan, J.A. 1973. Minimum mutation fits to a given tree. *Biometrics* 29: 53–65.

10. Lemieux C, Otis C, Turmel M. 2000. Ancestral chloroplast genome in *Mesostigma viride* reveals an early branch of green plant evolution. *Nature* 403:649–652.
11. Meidanis, J. and Munuera, E.G. 1996. A theory for the consecutive ones property. *Proceedings of WSP'97 - Third South American Workshop on String Processing*, N. Ziviani, R. Baeza-Yates, eds., Carleton University Press, 194–202.
12. Nadeau, J.H. and Taylor, B. 1984. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc Natl Acad Sci U S A.* 81:814–8.
13. Ohyama, K., Fukuzawa, H., Kohchi, T. and 13 co-authors. 1986. Chloroplast gene organization deduced from complete sequence of liverwort *Marchantia polymorpha* chloroplast DNA. *Nature* 322:572–574.
14. Parida, L. 2005. A PQ tree-based framework for reconstructing common ancestors under inversions and transpositions. *IBM Research Report RC 23837*.
15. Sankoff, D. and Blanchette, M. 1997. The median problem for breakpoints in comparative genomics. *Proceedings of the Third International Computing and Combinatorics Conference (COCOON 1997)*, T. Jiang and D.T. Lee, eds., *Lecture Notes in Computer Science* 1276, Springer, 251–263.
16. Sankoff, D., Bryant, D., Deneault, M., Lang, F. and Burger, G. 2000. Early eukaryote evolution based on mitochondrial gene order breakpoints. *Journal of Computational Biology* 7: 521–535.
17. Sankoff, D. and Rousseau, P. 1975. Locating the vertices of a Steiner tree in an arbitrary metric space. *Mathematical Programming* 9: 240–246.
18. Sankoff, D., Sundaram, G. and Kececiloglu, J. 1996. Steiner points in the space of genome rearrangements. *International Journal of the Foundations of Computer Science* 7:1–9.
19. Siepel, A. and Moret, B. 2001. Finding an optimal inversion median: experimental results. *Proceedings of the First Workshop on Algorithms in Bioinformatics (WABI 2001)*, *Lecture Notes in Computer Science* 2149, Springer, 189–203.
20. Tang, J., and Moret, B.M.E., 2003. Phylogenetic reconstruction from gene rearrangement data with unequal gene contents. *Proceedings of the Eighth Workshop on Algorithms and Data Structures (WADS 2003)*, *Lecture Notes in Computer Science* 2748, Springer, 37–46.
21. Telles, G.P. and Meidanis, J. 2005. Building PQR trees in almost-linear time. *Proceedings of GRACO 2005*, P. Feofiloff; C.M.H. de Figueiredo; Y. Wakabayashi, eds., *Electronic Notes in Discrete Mathematics* 19: 1-416.
22. Turmel M, Otis C, Lemieux C. 2002. The chloroplast and mitochondrial genome sequences of the charophyte *Chaetosphaeridium globosum*: insights into the timing of the events that restructured organelle DNAs within the green algal lineage that led to land plants. *Proc Natl Acad Sci U S A.* 99:11275–80.
23. Turmel, M., Otis, C. and Lemieux, C. 2005. The complete chloroplast DNA sequences of the charophycean green algae *Staurastrum* and *Zygnema* reveal that the chloroplast genome underwent extensive changes during the evolution of the *Zygnematales*. *BMC Biology* 3:22.
24. Turmel M, Otis C, Lemieux C. 2006. The chloroplast genome sequence of *Chara vulgaris* sheds new light into the closest green algal relatives of land plants. *Mol Biol Evol.* 23:1324–38.