On the reconstruction of ancient doubled circular genomes using minimum reversals

Nadia El-Mabrouk [1] **David Sankoff** [2]
mabrouk@iro.umontreal.ca sankoff@ere.umontreal.ca

[1] Département d'informatique et de recherche opérationelle, Université de Montréal, CP 6128 succursale Centre-Ville, Montréal,Québec H3C 3J7.
[2] Centre de recherches mathématiques, Université de Montréal, CP 6128 succursale Centre-Ville, Montréal,Québec H3C 3J7.

Abstract

We propose a model of the doubling of a bacterial genome followed by gene order rearrangement to explain present-day patterns of duplicated genes. On the hypothesis that inversion (reversal) is the predominant mechanism of rearrangement, we ask how to reconstruct the ancestral genome at the moment of genome duplication. We present a polynomial algorithm for finding such a genome that minimizes (within 2 reversals) the Hannenhalli-Pevzner formula for reversal distance from the modern genome. We illustrate by applying the algorithm to a set of duplicate genes in the Marchantia polymorpha *mitochondrial genome.*

1 Introduction

In 1985, Herdman [3] noticed that bacterial genome sizes clustered around multiples of 0.8Mb (i.e., 1.6Mb, 3.2Mb etc.). He suggested that many bacterial genomes are the product of ancient duplications. This has remained a controversial idea and often crops up in the literature, e.g. [4].

The gene order of modern-day bacteria is not strong evidence for or against ancient duplication. There are often pairs of regions which are similar in gene content and order, but these are too rare and scattered to be convincing proof of a genome-wide duplication. If this event did occur, it has since been almost totally obscured by loss or divergence (in sequence and function) of one or both of the copies of most gene pairs, and by extensive rearrangement of the gene order. Inversion of long or short chromosomal segments is often cited as one of the predominant mechanisms for gene order rearrangement in unichromosomal genomes.

This leads to the question: can we use genome rearrangement theory to reconstruct some or most of the original gene order at the time of genome duplication, based on conserved traces in the ordering of those duplicate genes still identifiable?

The Hannenhalli-Pevzner formula [2] calculates the exact number of reversals necessary to transform one gene order into another. This is not directly applicable, however, since we do not know *a priori* what the ancestral gene order was. In this paper, we show how to construct the ancestral order in such a way as to minimize (within two reversals) the Hannenhalli-Pevzner formula for the distance between the ancestor and its descendant, based on the gene order of the modern-day genome.

The method we use resembles in many ways the technique we developed previously to find the ancestral duplicated genome under a minimum translocations criterion in the multichromosomal eukaryote context [1], but the circular genome and the reversals operation studied here both simplify the statement of the problem and make its exact solution more difficult.

2 The problem

We define a **genome** as a circular string of signed (+ or −) terms (**genes**) from a set $\mathcal{B}$. (Though we will write genomes as linear strings, it is understood that the first term is adjacent to the right of the last gene.) A **rearranged duplicated genome** is a genome G where every gene in $\mathcal{B}$ appears exactly twice in G.

Example 1 $G = +a\ +b\ -c\ +b\ -d\ -e\ +a\ +c\ -d\ -e$.
G is a rearranged duplicated genome on the set of genes $\mathcal{B} = \{a, b, c, d, e\}$.

For a string $X = x_1 x_2 \cdots x_r$, denote by $-X$ the **reverse** string $-x_r - x_{r-1} \cdots - x_1$. A **reversal** (or **inversion**) transforms some proper substring of a genome into its reverse.

The ancestral genome of a rearranged duplicated genome was a string X containing one occurrence of each gene in $\mathcal{B}$. At the doubling event, two copies of this string were concatenated, to produce a **pristine duplicated genome** of form $X\ X$ or $X\ -X$. Thereafter, a series of reversals transformed the genome into its present configuration. Our problem is to estimate the ancestral genome using a minimum reversals criterion. We denote by $I(G)$ the minimum number of reversals necessary to transform G into a pristine duplicated genome.

3 The Hannenhalli graph

Let $\mathcal{B}$ be a set of genes. Let H_1 and H_2 be two circular genomes such that each of them contains all the genes of $\mathcal{B}$, where each gene appears exactly once. How many reversals does it take to transform H_1 into H_2? We denote by $I(H_1, H_2)$ this minimal number of reversals.

The Hannenhalli and Pevzner bicoloured **cycle graph** $\mathcal{G}_{12}$ of H_1 with respect to H_2 is as follows. If gene x_i in genome $H_1 = x_1 \cdots x_n$ has positive sign, replace it by the pair $x_i^t x_i^h$, and if it is negative, by $x_i^h x_i^t$. Then the vertices of $\mathcal{G}_{12}$ are just the x^t and the x^h for all x in $\mathcal{B}$. Any two vertices which are adjacent in H_1, other than x_i^t and x_i^h from the same x, are connected by a black edge, and any two adjacent in H_2, by a gray edge. Each vertex is incident to exactly one black and one gray edge, so that there is a unique decomposition of $\mathcal{G}_{12}$ into c_{12} disjoint cycles of alternating edge colours. Note that $c_{21} = c_{12} = c$ is maximized when $H_1 = H_2$, in which case each cycle has one black edge and one gray edge, and $c = |\mathcal{B}|$.

By the **size of a cycle** we mean the number of black edges it contains.

A key concept in the method is the **unoriented component**. A gray edge in a cycle of size > 1 is **oriented** if the reversal disrupting the two adjacent black edges, i.e.,

a adjacent to b in H_1, b adjacent to c in H_2, c adjacent to d in H_1
becomes
a adjacent to c in H_1, c adjacent to b in H_2, b adjacent to d in H_1,

replaces the cycle by two cycles. Otherwise, the edge is **unoriented**. Informally, an oriented gray edge links the vertex on the clockwise (or counterclockwise) side of a black edge to the vertex on the clockwise (or counterclockwise, respectively) side of another black edge, while an unoriented gray edge links two different sides of two black edges.

A cycle C is **oriented** if it contains at least one oriented gray edge. Otherwise it is **unoriented**. Two cycles containing gray edges that "cross", e.g., gene i adjacent to gene j in Cycle 1, gene k adjacent to gene t in Cycle 2 in G, but ordered i, k, j, t in H, are **connected**. A (connected) **component** of $\mathcal{G}_{12}$ is a subset of the cycles, built recursively from one cycle, at each step adding all the remaining cycles connected to any of those already in the construction. An **oriented component** has at least one oriented cycle and thus at least one oriented gray edge. Otherwise it is an **unoriented component**.

We say that component U **separates** two components U' and U'' if any edge we tried to draw from a vertex of U' to one of U'' would cut a gray edge of U. A **hurdle** is an unoriented component which does not separate any pair of components.

Hannenhalli and Pevzner show that the minimum number of reversals necessary to transform H_1 into H_2 is:

$$I(H_1, H_2) = b(\mathcal{G}_{12}) - c(\mathcal{G}_{12}) + h(\mathcal{G}_{12}) + fr(\mathcal{G}_{12}) \quad \mathbf{(1)}$$

where $b(\mathcal{G}_{12})$ is the number of black edges, $c(\mathcal{G}_{12})$ is the number of cycles, $h(\mathcal{G}_{12})$ is the number of hurdles of $\mathcal{G}_{12}$, and $fr(\mathcal{G}_{12})$ is a parameter which is 0 or 1 depending on the type of hurdles in $\mathcal{G}_{12}$ [2].

4 Preliminaries

To make use of the bicoloured graph structure for our problem, we introduce a distinction within each pair of identical genes in the rearranged duplicated genome G, labeling either one x_1 and the other x_2 for all x. Each x_j is replaced by x_j^t and x_j^h as in the Hannenhalli-Pevzner graph. Define:

$$\mathbf{V} = \{x_j^s\}_{\substack{x \in \mathcal{B} \\ j=1,2}}^{s \in \{h,t\}}.$$

We use the notation $\overline{1} = 2$, $\overline{2} = 1$, $\widetilde{t} = h$, $\widetilde{h} = t$. Note that $\overline{\overline{u}} = \widetilde{\widetilde{u}} = u$.

The **partial graph** $\mathcal{G}(\mathbf{V}, A)$ associated with G, has the edge set A of (black) undirected edges linking adjacent terms (other than x^t and x^h) in G. The partial graph associated with the genome in Example 1 is shown in Figure 1. To differentiate the two occurrences of each gene x, one is subscripted "1", its counterpart "2".

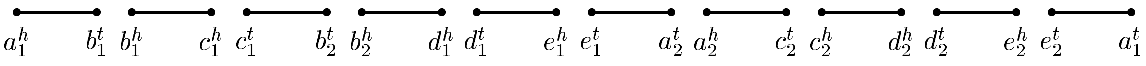

Figure 1: Partial graph corresponding to Example 1

The addition to the partial graph $\mathcal{G}(\mathbf{V}, A)$ of a set D of gray undirected edges, corresponding to some pristine duplicated genome H, produces a **completed graph** $\mathcal{G}_D(\mathbf{V}, A, D)$. Note that every vertex in $\mathbf{V}$ will then be incident to exactly one black edge and one gray edge.

Rewriting formula (1), we are required to find a pristine duplicated genome H, represented by a set D of gray edges, which minimizes the expression $|A| - c(\mathcal{G}_D) + h(\mathcal{G}_D) + f(\mathcal{G}_D)$, where $c(\mathcal{G}_D)$ is the number of cycles, $h(\mathcal{G}_D)$ is the number of hurdles associated with the graph $\mathcal{G}_D(\mathbf{V}, A, D)$, and $f(\mathcal{G}_D)$ is a parameter associated with the set of hurdles.

Lemma 1 follows directly from the above definitions.

Lemma 1 *Given a completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$, D contains exactly zero or two edges of form $(u, \overline{u})$, and if $(u, v) \in D$ then $(\overline{u}, \overline{v}) \in D$.*

As a first step, we will disregard hurdles, and concentrate on completing the partial graph $\mathcal{G}(\mathbf{V}, A)$ so as to maximize the number of cycles, producing a **maximal completed graph**.

4.1 Decomposition into subgraphs

We define the set of **natural subgraphs** of $\mathcal{G}(\mathbf{V}, A)$ as follows. (cf. Figure 2.)

Definition : Let $e = (u, v) \in A$. Define A_e recursively by $(u, v) \in A_e$, and if $(x, y) \in A_e$ then both the edge of A adjacent to $\overline{x}$ and the edge of A adjacent to $\overline{y}$ are also in A_e.

Let $\mathbf{V}_e$ be the subset of $\mathbf{V}$ made up of vertices incident to the edges in A_e. Then $\mathcal{G}_e(\mathbf{V}_e, A_e)$ is the natural subgraph (of size $|A_e|$) of $\mathcal{G}(\mathbf{V}, A)$ generated by e. Note that if $f \in A_e$, then $A_f = A_e$

We then divide the natural subgraphs of $\mathcal{G}(\mathbf{V}, A)$ into the $\mathcal{CE}$, containing those subgraphs of even size, and GCO, containing those of odd size.

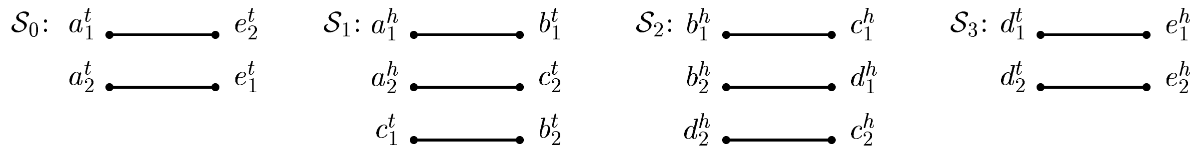

Figure 2: The natural subgraphs for Example 1. $\mathcal{S}_1$ and $\mathcal{S}_2$ are in GCO, $\mathcal{S}_0$ and $\mathcal{S}_3$ are in $\mathcal{CE}$.

The set A contains $2|\mathcal{B}|$ edges, and subgraphs in $\mathcal{CE}$ contain an even number of edges. Then GCO must also contain an even number of edges, and thus an even number of subgraphs. We can then arbitrarily pair off subgraphs in GCO and amalgamate each pair. The set of larger subgraphs thus formed is denoted $\mathcal{CO}$. The subgraphs in $\mathcal{CE} \cup \mathcal{CO}$ are called **supernatural subgraphs**.

Example 2 *Consider the natural subgraphs of Figure 2. Let $\mathcal{S}_{12}$ be the supernatural subgraph of $\mathcal{CO}$ obtained by amalgamating $\mathcal{S}_1$ and $\mathcal{S}_2$. Then the set $\{\mathcal{S}_0, \mathcal{S}_{12}, \mathcal{S}_3\}$ is a decomposition of $\mathcal{G}(\mathbf{V}, A)$ into supernatural subgraphs.*

5 Maximizing the number of cycles

Let α be the parameter defined as follows: if $\mathcal{CO}$ is empty then $\alpha = |\mathcal{CE}|$, otherwise $\alpha = |\mathcal{CE}| + 1$. Using the methods of Theorem 2 in [1], we can prove:

Theorem 1 *Let c_{max} be the number of cycles of a maximal completed graph of $\mathcal{G}(\mathbf{V}, A)$. Then:*

$$c_{max} \leq \frac{|A|}{2} + \alpha$$

We now complete the supernatural subgraphs, obtaining c_{max} cycles. The main characteristic of the algorithm is that each gray edge should link two vertices of the same supernatural subgraph. At each step, we denote by F the set of fragments of the genome H resulting from the preceding steps. At the outset, F is made up of the **unitary fragments**, which are made up of pairs of vertices, $a^t a^h$, for all $a \in \mathcal{B}$. As the construction proceeds, whenever a gray edge (x, y) is created, the fragment containing x and the one containing y are joined together. A **long fragment** is one that is not unitary. A **terminal fragment** is a circular one. Note that the only vertices of a non terminal fragment not adjacent to gray edges are its two endpoints.

The general strategy of the algorithm is as follows:

- Consider two unitary fragments x and $\overline{x}$;
- Extend these two fragments, obtaining two long fragments of form X and $\overline{X}$.
- If $\mathcal{CO} \neq \emptyset$, at some point in the algorithm, a gray edge of form $(x, \overline{x})$ will link two long fragments to form a single long fragment of form $X\ -\overline{X}$. The algorithm continues to extend this long fragment, at each step adding a vertex at one end and its inverse at the other.
- The final step of the algorithm links two long fragments of forms X, $\overline{X}$ to obtain a pristine duplicated genome ($X\overline{X}$ or $X\ -\overline{X}$), or closes the circle of a single long fragment of form $X\ -\overline{X}$, thus producing a pristine duplicated genome.

Let x, y be two distinct vertices, not yet linked by a gray edge. To be able to construct the edge (x, y), we must have the three next conditions satisfied:

1. $y \neq \widetilde{\overline{x}}$;

2. If two gray edges of form $(z, \overline{z})$ have already been constructed, then $y \neq \overline{x}$;

3. If adding (x, y) produces a terminal fragment, then F contains no other fragments.

For condition 1 to be satisfied, it suffices to avoid, at the outset, a vertex x adjacent to a black edge $(x, \widetilde{\overline{x}})$. At each step, the algorithm constructs a gray edge adjacent to the last vertex of the fragment obtained in the preceding step. Suppose that at a certain step of the algorithm, the gray edge (x, y) is constructed. If $y \neq \overline{x}$, the next vertex to be considered is $\widetilde{y}$; otherwise, there must be a single long fragment with endpoints z and $\overline{z}$, and z is the next vertex to be considered. Let x^n be the next vertex to be considered after having constructed (x, y) ($\widetilde{y}$ in the first case and z in the second case). So that the algorithm is not stopped prematurely after having constructed the edge (x, y), we must make sure that it is possible to construct an edge from the endpoint x^n. In other words, we must take care, in constructing a gray edge (x, y), not to get into an impossible configuration, characterized by one of the following four conditions. A configuration which avoids all of these properties is termed **possible**.

Propertie I. There are both unitary and terminal fragments.

Propertie II. The graph containing $z = x^n$ contains no more than the four unitary fragments z, $\overline{z}$, t, $\overline{t}$, the gray edge (z, t) forms a terminal fragment, but there are other unitary fragments in other supernatural subgraphs.

Propertie III. The graph containing $z = x^n$ contains no more than the two unitary fragments z, $\overline{z}$, the gray edge $(z, \overline{z})$ forms a terminal fragment, but there are other unitary fragments in other supernatural subgraphs.

Propriété IV. $z = x^n$ is on a path for which $\overline{z}$ is the other endpoint, and the graph containing this path also contains another path C_2 with endpoints t and $\overline{t}$. Moreover, one of the two next situations is satisfied: (1) if an edge of form $(a, \overline{a})$ has already been constructed, then the construction of (z, t) leads to Properties I, II or III; (2) if not, the construction of $(z, \overline{z})$ leads to Property III.

Clearly, if $y \neq \overline{x}$ and (x, y) do not lead to an impossible situation, then (x, y) satisfies condition (3). A pair of vertices (x, y) such that $y \neq \overline{x}$ is said to be **impossible** if it creates an impossible situation, and **possible** otherwise.

Let x be a vertex of $\mathbf{V}$ not yet linked by a gray edge. Then in the bicoloured graph obtained at this step, x is one of the two endpoints of a path C (made up of a succession of black and gray edges). Call the second endpoint of this path x^c. Note that $\overline{x}^c$ is, in general, different from x^c.

Figure 3 describes an algorithm for constructing a completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$. In this description, we not repeat the fact each time a gray edge (x, y) is created, this implies the creation of $(\overline{x}, \overline{y})$.

Lemma 2 *Algorithm dedouble is correct, i.e. it produces a completed graph.*

Theorem 2 *The completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$ constructed by dedouble contains $c(\mathcal{G}) = \alpha + \frac{|A|}{2}$ cycles. Therefore, by theoreme 1, it is maximal.*

(The full version of this paper contains complete proofs.)

Example 3 *Consider genome G in Example 1, and the decomposition of its graph $\mathcal{G}(\mathbf{V}, A)$ into the supernatural subgraphs of example 2. Here $|A| = 10, \alpha = 3$, and thus $c = \alpha + \frac{A}{2} = 8$. Figure 4 depicts the completed graph produced. The corresponding pristine duplicated genome is*

$$H = -a_1\ +e_2\ +d_2\ -c_2\ +b_1\ -b_2\ +c_1\ -d_1\ -e_1\ +a_2$$

Algorithm dedouble-circular:
Choose a vertex x not adjacent to a black edge of form $(x, \overline{\overline{x}})$;
While $\mathcal{G}$ contains unitary fragments, do
  If (x, x^c) is possible
    Construct the edge (x, x^c);
    If $x^c \neq \overline{x}$
      Set $x = \widetilde{x^c}$;
    Otherwise, let z and $\overline{z}$ be the two endpoints of the only long fragment of F;
      Set $x = \widetilde{z}$;
  Otherwise,
    If $x^c \neq \overline{x}$
      Construct edge $(\overline{x}, \overline{x}^c)$
      Set $x = \widetilde{\overline{x}^c}$;
    Otherwise, the supernatural graph $\mathcal{G}'$ containing x is in $\mathcal{CO}$. Let $\mathcal{G}'_1$ and $\mathcal{G}'_2$ be the two constituent natural graphs of $\mathcal{G}'$, such that $x \in \mathcal{G}'_1$
      If there are only two vertices y and $\overline{y}$ in $\mathcal{G}'_2$ not yet linked by gray edges
        Construct the edge (x, y);
        Set $x = \widetilde{y}$;
      Otherwise, there must be vertices y, $\overline{y}$, z and $\overline{z}$ in $\mathcal{G}'_2$ not yet linked by gray edges
        If (x, y) is possible
          Construct the edge (x, y);
          Set $x = \widetilde{y}$;
        Otherwise
          Construct edge (x, z);
          Set $x = \widetilde{z}$;

Figure 3: Algorithm for constructing a maximal completed graph.

6 Hurdles

Recall that our goal is to complete the partial graph $\mathcal{G}(\mathbf{V}, A)$ in such a way so as to minimize formula (1). Up to now, we have been only trying to maximize the number of cycles in a completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$. Now we turn our attention to minimizing the number of hurdles.

A hurdle is a particular kind of **large** (i.e., of size > 1) component. In a cycle graph, a large component C can be identified by a fragment $I = [u_l, u_r]$ where u_l (or u_r) is the extreme vertex of C in the counterclockwise (or clockwise, respectively) direction, such that no gray edge links a vertex of I to a vertex outside of I, and such that at least one cycle of I is of size > 1. We call such an interval **the interval of the component** C. We also identify a **minimal component** as a large component whose interval contains no other component.

Definition : Let $\mathcal{G}_D(\mathbf{V}, A, D)$ be a completed graph, G the initial genome and H the pristine duplicated genome represented by the set D of gray edges. The sequence $S = u_1 u_2 \cdots u_{p-1} u_p$ is a **subpermutation** of $\mathcal{G}_D(\mathbf{V}, A, D)$ (or of the pair (G, H)) if S is a subsequence of G, and if there is a permutation P, such that $T = P(S) = u_1 v_2 \cdots v_{p-1} u_p$ is also a subsequence of H, where $v_2 \neq u_2$ and $v_{p-1} \neq u_{p-1}$.

A **minimal subpermutation** of $\mathcal{G}_D(\mathbf{V}, A, D)$ contains no other subpermutation of $\mathcal{G}_D(\mathbf{V}, A, D)$.

There is a bijection between the subpermutations and the components of size > 1 of $\mathcal{G}_D(\mathbf{V}, A, D)$. Specifically, let S be a subpermutation of $\mathcal{G}_D(\mathbf{V}, A, D)$, $\Pi = \{\pi_1, \cdots \pi_p\}$ the set of components con-

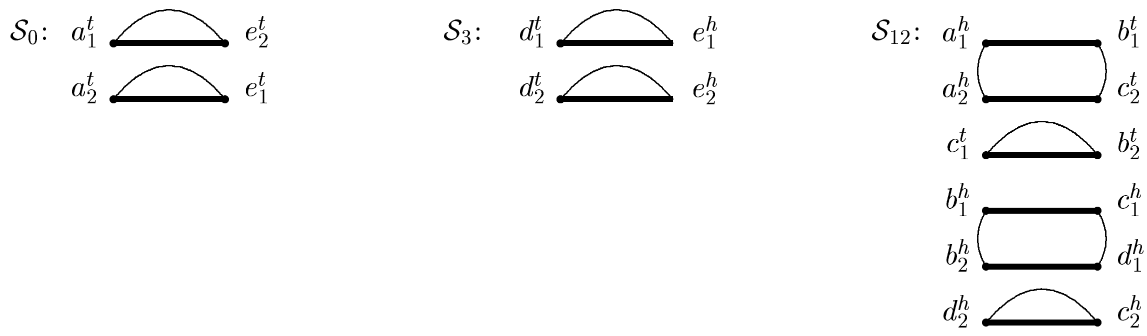

Figure 4: Completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$ found by algorithm **dedouble**.

taining the vertices of S, and $\mathcal{V} = \{\mathbf{V_1}, \cdots, \mathbf{V_p}\}$ where for all i, $\mathbf{V_i}$ is the set of vertices of π_i. Then:

- S_i is an inner subpermutation of S (possibly S itself) if and only if S_i corresponds to an interval of a component π_i of Π. We call this component **the component of the subpermutation** S_i.
- S_i is a minimal inner subpermutation of S if and only if S_i corresponds to an interval of a minimal component of Π.

Example 4 *Consider the two following genomes and the corresponding completed graph (Figure 5).*

$$G = +a_1 \ +b_1 \ +c_1 \ +d_1 \ +e_1 \ +d_2 \ -f_1 \ -e_2 \ -f_2 \ +a_2 \ -b_2 \ +c_2$$

$$H = +a_1 \ +b_1 \ +c_1 \ +d_1 \ +e_1 \ +f_2 \ -f_1 \ -e_2 \ -d_2 \ -c_2 \ -b_2 \ -a_2$$

Each of the connected components (CC) of this graph is made up of a single cycle. There are three CC made up of a cycle of size > 1: $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$.
$\mathcal{C}_1$ is the CC of the subpermutation $S_1 = +e_1 \ +d_2 \ -f_1 \ -e_2 \ -f_2 \ +a_2 \ -b_2 \ +c_2 \ +a_1$.
$\mathcal{C}_2$ is the CC of the subpermutation $S_2 = +d_2 \ -f_1 \ -e_2 \ -f_2$.
$\mathcal{C}_3$ is the CC of the subpermutation $S_3 = +a_2 \ -b_2 \ +c_2$.
The only two minimal subpermutations are S_2 and S_3.

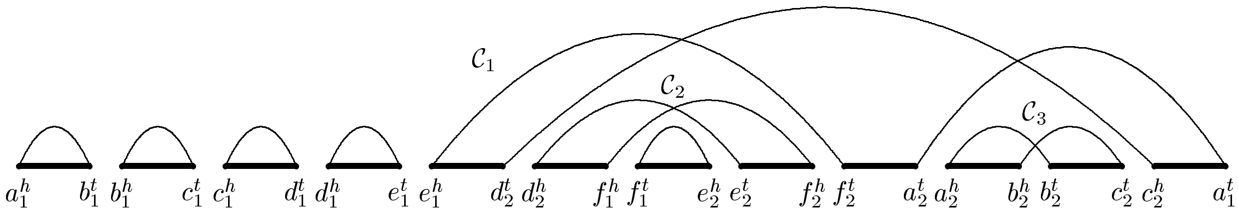

Figure 5: Completed graph corresponding to genomes G and H

Hurdles correspond to a certain kind of subpermutation. To minimize the number of hurdles we must then find the completed graph with the minimum number of these subpermutations. To do this we will make use of an additional notion:

Definition : A **local subpermutation of the genome** G is a subsequence $S = u_1 u_2 \cdots u_{p-1} u_p$ of G such that there exists another subsequence of G of form $\overline{S} = \overline{u_1} P(\overline{u_2}, \cdots, \overline{u_{p-1}}) \overline{u_p}$, where P is a permutation other than the identity. We call $\overline{S}$ the **complementary sequence** of S.

We say that a local subpermutation S is **minimal** if it contains no other local subpermutation.

Example 5 *Let the genome* $G = +a_1\ +b_1\ +c_1\ +d_1\ +e_1\ -d_2\ +b_1\ +c_1\ -a_2\ +e_2$.
The subsequence $S = +a_1 +b_1 +c_1 +d_1$ *is a local subpermutation of* G.
In the genome G *of Example 4, the subsequence* $+a_1 +b_1 +c_1$ *is a local subpermutation of* G.

Consider now the non-local subpermutations. We distinguish between those which do not contain both x and $\overline{x}$ for any vertex x, which we call **normal** and those that do, the **special** ones.

We now show how to correct *dedouble* so that when applied to a genome containing no local subpermutations it produces a maximal completed graph containing no normal subpermutations.

Let $\mathcal{G}_D(\mathbf{V}, A, D)$ be a maximal completed graph produced by *dedouble* and $S = x_1 \cdots x_n$ a normal subpermutation of $\mathcal{G}_D(\mathbf{V}, A, D)$. The following procedure applies to the subpermutation S.

Procedure spoil-SP$(x_1 \cdots x_n)$
Remove all the edges of D adjacent to the vertices of $\{x_1, \cdots, x_n, \overline{x_1}, \cdots, \overline{x_n}\}$;
Construct edges (x_k, x_{k+1}) and $(\overline{x_k}, \overline{x_{k+1}})$ for all k, $1 \leq k < n$.

Lemma 3 *Suppose that the completed graph* $\mathcal{G}_D(\mathbf{V}, A, D)$ *produced by* dedouble *contains a subpermutation* $S = x_1 x_2 \cdots x_{n-1} x_n$. *If* S *is not a local subpermutation of* G, *then* spoil-SP*(*$x_1 \cdots x_n$*) gives rise to a completed graph* $\mathcal{G}_{D'}(\mathbf{V}, A, D')$ *containing at least the same number of cycles as* $\mathcal{G}_D(\mathbf{V}, A, D)$, *and one less subpermutation.*

Consider $\mathcal{G}_D(\mathbf{V}, A, D)$ the maximal completed graph produced by *dedouble*. Let $\mathcal{S}$ be the set of subpermutations of $\mathcal{G}_D(\mathbf{V}, A, D)$ which are not local.

Procedure spoil-SP
For all normal $S \in \mathcal{S}$
spoil-SP(S).

As a corollary to Lemma 3, we have.

Corollary 1 *For* G *a genome with no local subpermutation, the completed graph produced by* dedouble *followed by* spoil-SP, *contains no normal subpermutations.*

Example 6 *Let genome* $G = +a_1\ +b_1\ +c_1\ +b_2\ +d_1\ +e_1\ -a_2\ +c_2\ +e_2\ +d_2$.
G contains no local subpermutation. The partial graph $\mathcal{G}(\mathbf{V}, A)$ *gives rise to a single supernatural graph belonging to* $\mathcal{CE}$. *Thus* $\alpha = 1$ *and since* $|A| = 10$, *a completed graph of* $\mathcal{G}(\mathbf{V}, A)$ *contains a maximum of 6 cycles.*

Consider the maximal completed graph portrayed in Figure 6.

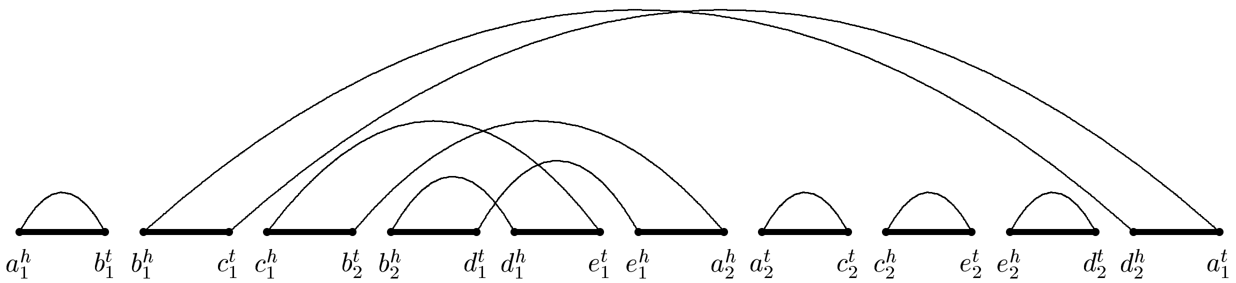

Figure 6: A maximal completed graph containing a normal subpermutation

The resulting genome is $H_1 = +a_1\ +b_1\ -d_2\ -e_2\ -c_2\ +a_2\ +b_2\ -d_1\ -e_1\ -c_1$.
The subsequence $S = +c_1 +b_2 +d_1 +e_1 -a_2$ *of* G *is a normal subpermutation of* (G, H_1).

Figure 7 represents the completed graph obtained through spoil-SP *applied to the completed graph of Figure 6. This graph is also a maximal completed graph since it also contains 6 cycles. The resulting genome is* $H_2 = +a_1\ -e_2\ -d_2\ -b_1\ -c_2\ +a_2\ -e_1\ -d_1\ -b_2\ -c_1$.
This new graph contains no normal subpermutation and a single special one.

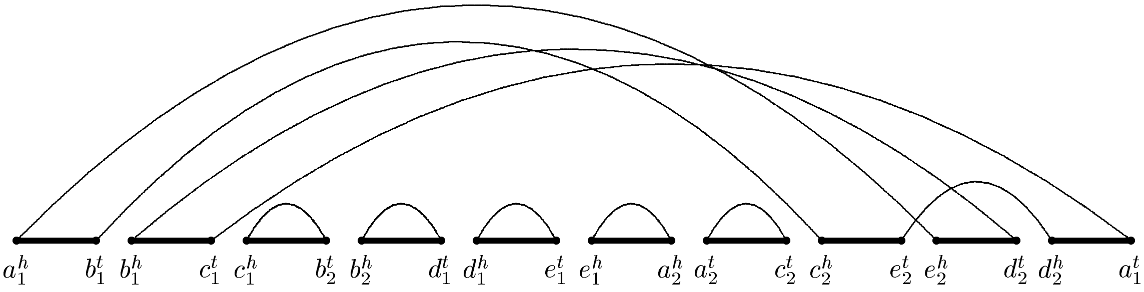

Figure 7: A maximal completed graph without any normal subpermutation

In the remainder of this paper it will be implicit that the procedure *spoil-SP* is incorporated at the end of the algorithm *dedouble*.

We turn next to the case where G contains local subpermutations.

Lemma 4 *Suppose G contains a local subpermutation $S = x_1 \cdots x_n$. Suppose the completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$ contains no subpermutation made up of the vertices $\{x_1, \cdots, x_n\}$. If c_{max} is the maximal number of cycles of a completed graph of $\mathcal{G}(\mathbf{V}, A)$ and c is the number of cycles of $\mathcal{G}_D(\mathbf{V}, A, D)$, $c \leq c_{max} - 2$.*

Let S be a local subpermutation, and let $\overline{S}$ be the complementary sequence of S. In order to obtain suitable cycles, we establish a constraint on the order of the edges in the supernatural graphs containing the vertices of $S \cup \overline{S}$: they all must start with a black edge from $\overline{S}$. Through the use of this order, it is easy to prove that *dedouble* constructs only cycles of size 1 for $\overline{S}$, and at least one cycle of size > 1 for S. In addition, S constitutes a subpermutation of the graph thus obtained.

Let $\Pi = \{\pi_1, \cdots, \pi_r\}$ be the set of components containing the vertices of S, produced by *dedouble*, and $\mathcal{V} = \{\mathbf{V_1}, \cdots, \mathbf{V_p}\}$ where for all i, $\mathbf{V_i}$ is the set of vertices of π_i.

In order to consider the components which may form hurdles, we introduce another definition. Let $\mathcal{U} = \{u_1, \cdots, u_p\}$ be a subset of $\mathcal{B}$, and $\overline{\mathcal{U}} = \{\overline{u_1}, \cdots, \overline{u_p}\}$. We say that $\mathcal{U}$ is **unoriented** if genes u_i and $\overline{u_i}$ either have the same sign in G, or opposite signs, for all i. The same criterion can be used to define oriented and unoriented sets of vertices.

Consider a component π_i of Π where $\mathbf{V_i}$ is its vertex set.

Lemma 5 *π_i is oriented if and only $\mathbf{V_i}$ is.*

We say that a **local subpermutation is oriented** if the set of vertices in its component is oriented, and is unoriented otherwise. Hurdles produced by *dedouble* and resulting from local subpermutations thus correspond to minimal unoriented subpermutations, which we call **local minimal subpermutations** and to at most one other subpermutation. Let $S = x_1 \cdots x_n$ be a local subpermutation. The **outer subpermutation** of S is the largest subpermutation S_e contained in S satisfying:

- The component π_e of S_e is unoriented.
- S_e is not minimal and the interval of S_e contains all the local minimal subpermutations of S.
- S_e does not separate two local minimal subpermutations.

Example 7 *Suppose the genome G contains the local subpermutation $S_1 = a_1\ c_1\ e_1\ d_1\ f_1\ h_1\ g_1\ i_1\ b_1\ j_1$, with complement $\overline{S_1} = a_2\ b_2\ c_2\ d_2\ e_2\ f_2\ g_2\ h_2\ i_2\ j_2$. The components of S_1 and the inner subpermutations of S_1 are depicted in Figure 8. The two components $\mathcal{C}_2$ and $\mathcal{C}_3$ correspond to components of the two local minimal subpermutations of S. $\mathcal{C}_1$ is the component of S_1. It is unoriented and does not separate two local minimal subpermutations. S_1 is thus an outer subpermutation.*

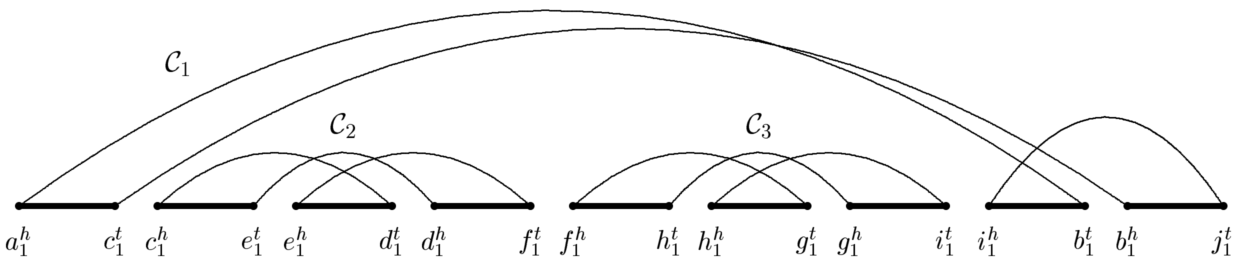

Figure 8: Inner subpermutations corresponding to S_1.

We say that a local subpermutation S is **maximal** if all the local minimal subpermutations of G are inner subpermutations of S and if there exists an outer subpermutation of S.

A **bad subpermutation** is a local subpermutation which is either minimal or maximal. By $\mathbf{brs(G)}$ we denote the number of bad subpermutations of G.

By Corollary 1 and Lemmas 4 and 5, the completed graph $\mathcal{G}_D(\mathbf{V}, A, D)$ of $\mathcal{G}(\mathbf{V}, A)$ produced by *dedouble* contains exactly $\mathbf{brs(G)}$ hurdles corresponding to bad subpermutations of G. In addition, there may be at most two more **special** hurdles due to the special subpermutations defined in our discussion of non-local subpermutations above.

We require one more lemma.

Lemma 6 *Suppose G contains an unoriented local subpermutation S. Let π be the component of S. Any maximal completed graph must contain an unoriented component made up of the vertices of π.*

Consider the parameter $f(G)$ which is 1 if the hurdles determined by the bad subpermutations of G form a fortress [2], and 0 otherwise.

Theorem 3 *Let $\mathcal{G}_D(\mathbf{V}, A, D)$ be the completed graph produced by* dedouble*, and H the resulting pristine duplicate genome. Then*

$$\frac{|A|}{2} - \alpha + brs(G) + f(G) \leq I(G, H) \leq \frac{|A|}{2} - \alpha + brs(G) + f(G) + 2$$

In addition: $$\frac{|A|}{2} - \alpha + brs(G) + f(G) \leq I(G) \leq \frac{|A|}{2} - \alpha + brs(G) + f(G) + 2.$$

Following Theorem 3, we have:

Corollary 2 *Let $\mathcal{G}_D(\mathbf{V}, A, D)$ be the completed graph of $\mathcal{G}(\mathbf{V}, A)$ produced by* dedouble*. Then if $\mathcal{G}_D(\mathbf{V}, A, D)$ does not contain any special hurdle, then*

$$I(G) = \frac{|A|}{2} - \alpha + brs(G) + f(G)$$

Example 8 *Consider genome G of Example 1. The supernatural graphs of G and the completion of these graphs by* dedouble *is depicted in Figure 4. The completed graph obtained is given in Figure 9.*

The number of cycles in this graph is $c(G) = 8$, $|A| = 10$, $brs(G) = 0$ and $f(G) = 0$. It does not contain any hurdle. Thus the minimum number of reversals necessary to transform genome G into a pristine duplicated genome is $I(G) = 10 - 8 + 0 + 0 = 2$.

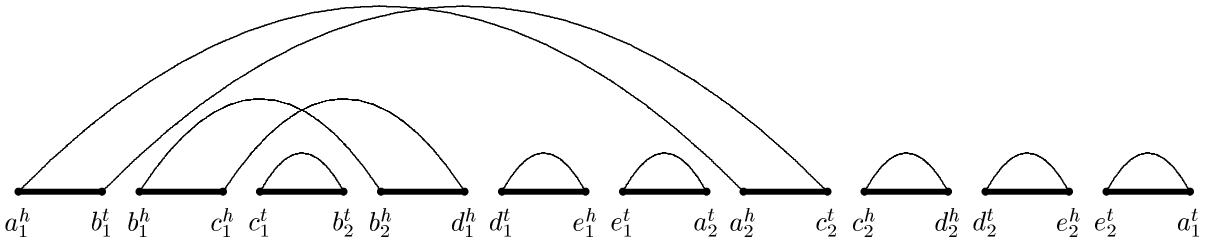

Figure 9: The completed graph obtained by algorithm dedouble, for genome G of example 1.

7 An application

The mitochondrial genome of the liverwort plant *Marchantia polymorpha* is rather unusual in manifesting many of its genes in two or three copies [5]. It is very unlikely that these arose from genome doubling, since this would not account for the numerous triplicates, nor is it consistent with comparative data on mitochondrial genomes. Nevertheless, it provides a convenient small example to test our method. A somewhat artificial map was extracted from the Genbank entry, deleting all singleton genes and one gene from each triplet (the two genes furthest apart were saved from each triplet). This led to a "rearranged duplicate genome" with 25 pairs of genes. A single supernatural subgraph in $\mathcal{CE}$ emerged from the analysis. This produced an estimate of 25 inversions (and a lower bound of 24), which is what one would expect from a random distribution of the duplicate genes on the genome. Any trace of genome duplication, were this even biologically plausible, has been obscured.

Acknowledgments

Research supported by grants to the authors from the Natural Sciences and Engineering Research Council of Canada. DS is a Fellow, and NEM a Scholar, in the Evolutionary Biology Program of the Canadian Institute for Advanced Research. Thanks to Lawrence Moran for introducing us to the literature on bacterial genome doubling and to Mélanie Deneault for finding the *Marchantia* data.

References

[1] El-Mabrouk, N., Bryant, B., Sankoff, D., Reconstructing the pre-doubling genome. In Istrail, S., Pevzner ,P. and Waterman, M. (eds) *Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB'99),* ACM press, 154–163, 1999.

[2] Hannenhalli, S. and Pevzner, P.A., Transforming cabbage into turnip. (polynomial algorithm for sorting signed permutations by reversals). In: *Proceedings of the 27th Annual ACM-SIAM Symposium on the Theory of Computing*, 178–189, 1995.

[3] Herdman, M. The evolution of bacterial genomes. In Cavalier-Smith, T. (ed) *The Evolution of Genome Size.* John Wiley & Sons, New York, 1985.

[4] Kunisawa, T. *Journal of Molecular Evolution* 40:585–593, 1995.

[5] Oda, K., Yamato ,K., Ohta, E., Nakamura, Y., Takemura, M., Nozato, N., Kohchi, T., Ogura, Y., Kanegae, T., Akashi, K. and Ohyama, K., Gene organization deduced from the complete sequence of liverwort Marchantia polymorpha mitochondrial DNA. A primitive form of plant mitochondrial genome. *Journal of Molecular Biology* 223:1–7,1992.