Conservation of Combinatorial Structures in Evolution Scenarios

Anne Bergeron, Cedric Chauve
LaCIM and Département d'Informatique
Université du Québec à Montréal

Sèverine Bérard
LIRMM, Université Montpellier II

To appear in the proceedings of the RECOMB Satellite Workshop on Comparative Genomics 2004.

Conservation of Combinatorial Structures in Evolution Scenarios

Sèverine Bérard[1], Anne Bergeron[2], and Cedric Chauve[2]

[1] LIRMM, Montpellier, France. E-mail: berard@lirmm.fr
[2] LaCIM, Université du Québec à Montréal, Canada. E-mail: [anne,chauve]@lacim.uqam.ca

Abstract. This paper investigates the problem of conservation of combinatorial structures in genome rearrangement scenarios. We give a characterization of a class of scenarios that conserve all common intervals, called commuting scenarios, and a characterization of permutations for which commuting scenarios exist. We show that measuring conservation of common intervals can be useful tool in assessing the quality of rearrangement scenarios, by investigating in detail three specific scenarios involving the mouse, rat and human X chromosomes.

1 Introduction

The reconstruction of evolution scenarios based on genome rearrangements has proven to be a powerful tool in understanding the evolution of close species, especially mammals. For example, in the last two years, several very interesting evolution scenarios have been proposed between the mouse and the human [15], and the between human, the mouse and the newly sequenced Norway rat [7, 10], using the MGR and GRIMM software [6, 17].

MGR relies heavily on *sorting signed permutations by inversions* and the related *median* problem [14]. However, the number of parsimonious sequences of inversions can be exponential [3]. It is then natural to ask for some additional criteria that can help to select putative scenarios. We are interested in scenarios that do not break combinatorial structures, defined in terms of genomic segments, that are conserved in both chromosomes. Indeed, if two genomes share a common feature, it is likely that their common ancestor did too, which makes evolution scenarios that conserve this feature interesting.

In this work, the combinatorial structures that we consider are *common intervals* [18, 12]. We give a characterization of a class of evolution scenarios between two chromosomes that are both parsimonious and do not break any interval of genomic segments that is common to the two chromosomes. We call this class of scenarios *commuting scenarios*, and we describe a linear time algorithm to decide if a commuting scenario exists, and compute it if it does exist.

Sections 2 and 3 present the basic concepts and definitions. In Section 4, we discuss results on human, mouse and rat chromosomes X [10], highlighting the role of common intervals in assessing the quality of evolution scenarios. The main theoretical results are presented in Section 5.

2 Rearrangement Scenarios

A *signed permutation* is a permutation on the set of integers $\{0, 1, 2, \ldots, n\}$ in which each element has a sign, positive or negative. An *inversion* of an interval of a signed permutation inverts the order of the elements of the interval, while changing their signs. In the following, we will assume that genomes are modeled by signed permutations, and that rearrangement operations are restricted to inversions. A *rearrangement scenario* between two or more genomes is given by an unrooted tree whose nodes are labeled by permutations and such that each of the given genomes labels a leaf, and the permutations labeling two adjacent nodes differ by one inversion.

The number of vertices of the tree is one more the number of rearrangements of the scenario. A scenario with a minimum number of rearrangements is called an *optimal* scenario. For example, given the three permutations $G_1 = (1\ 2\ 3\ 4\ 5\ 6)$, $G_2 = (1\ \overline{3}\ \overline{2}\ 5\ 4\ 6)$, $G_3 = (1\ \overline{5}\ \overline{2}\ \overline{4}\ 3\ 6)$, two rearrangement scenarios for G_1, G_2 and G_3 are proposed in Fig. 1, each of them having 6 rearrangements.

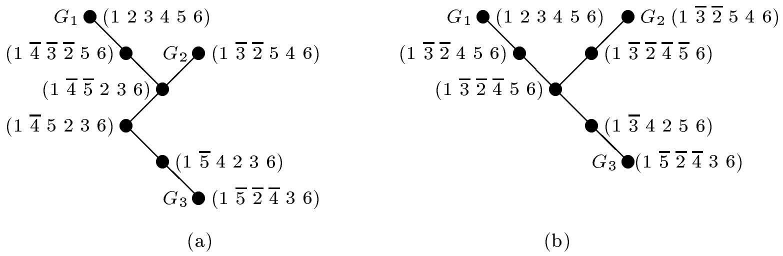

Fig. 1. Two rearrangement scenarios between permutations G_1, G_2 and G_3.

With two permutations, there exist polynomial algorithms to compute an optimal scenario [11, 13, 2, 16], but the problem becomes NP-hard for more than two permutations [8], although good heuristics are available [14].

Usually, there is more than one optimal scenario, even with different tree topologies. For example, Fig 1(b) gives an alternate scenario for genomes G_1, G_2 and G_3, that yields a different *median*, or *common ancestor*, for the three species. Is one scenario "better" than the other? When dealing with real genomes, only further insight from biology and evolution history will allow to completely settle this question. However, we will show that tracking some combinatorial structures along different scenarios can help to partially assess the quality of a scenario. For example, using data from chromosomes X of the mouse, human and rat [10], we were able to detect a major difference between two mouse assemblies – a transposition of two blocks of more than 300k base pairs.

3 Common Intervals and Commuting Inversions

A *point* $p \cdot q$ in a permutation is defined by a pair of consecutive elements in the permutation. When a point is of the form $i \cdot i + 1$, or $-(i + 1) \cdot -i$, it is called an *adjacency*, otherwise it is called a *breakpoint*.

An interval of a permutation is defined either by giving its *endpoints*, or by giving the set of its (unsigned) elements $\{|p_i|, \ldots, |p_j|\}$. A non-empty interval of the identity permutation can also be specified by giving its first and last element, such as $[i..j]$, in which case it is then understood that all elements between i and j belong to the interval.

The notion of *common interval* was studied among others in [12] in order to model the fact that a group of genes can be rearranged in a genome but still remain connected.

Definition 1. *A* common interval *of two signed permutations P and Q is a set of two or more integers that is an interval in both permutations.*

For example, the common intervals of permutations $G_2 = (1\ \overline{3}\ \overline{2}\ 5\ 4\ 6)$ and $G_3 = (1\ \overline{5}\ \overline{2}\ \overline{4}\ 3\ 6)$ are $\{2,5\}$, $\{2,4,5\}$, $\{2,3,4,5\}$, $\{1,2,3,4,5\}$, $\{2,3,4,5,6\}$, and $\{1,2,3,4,5,6\}$.

Definition 2. *Two distinct sets of integers A and B are said to* commute *if they trivially intersect, that is, $A \subset B$, or $B \subset A$, or $A \cap B = \emptyset$.*

The above definition, together with the fact that both intervals and inversions can be represented as sets of integers, is central in the links we establish between inversions, commutation and conservation of intervals. Indeed, an inversion that commutes with an interval does not change the composition of this interval, while it may change the order of the elements within the interval.

Definition 3. *The* score *of a scenario between two signed permutations P and Q is the ratio of inversions that commute with all common intervals of P and Q over the number of inversions in the scenario.*

For example, each of the rearrangement scenarios of Fig. 1 induces three scenarios with two permutations. The scores are given in Table 1, and show that the second scenario is slightly better, in terms of conservation, than the first.

	Scenario (a)	Scenario (b)
G_1 to G_2	0/3	4/4
G_1 to G_3	2/5	2/4
G_2 to G_3	2/4	2/4
Total	4/12	8/12

Table 1. Conservation scores of the two evolution scenarios of Fig. 1.

Efficient computation of scores is based on the notion of *irreducible intervals*. A common interval I between P and Q is an irreducible interval if there is an adjacency of P contained in I, and I is the smallest common interval between P and Q that contains this adjacency. For example, irreducible intervals between G_1 and G_2 of Fig. 1 are: $\{1,2,3\}$ (for adjacency $1 \cdot 2$), $\{2,3\}$ (for adjacency $2 \cdot 3$), $\{2,3,4,5\}$ (for adjacency $3 \cdot 4$), $\{4,5\}$ (for adjacency $4 \cdot 5$), and $\{4,5,6\}$ (for adjacency $5 \cdot 6$). Any common interval is the union of a sequence of irreducible intervals such that two consecutive intervals intersect. This implies:

Proposition 1. *An inversion breaks a common interval if and only if it breaks an irreducible interval.*

Since, there are at most n irreducible intervals between two signed permutations of $\{1, 2, \ldots, n\}$, and they can be identified in linear time [12], Proposition 1 implies that computing scores can be done efficiently.

4 Reconstructing the Ancestral Chromosome X

The scenario proposed in [10], that attempts to reconstruct a putative chromosome X for the common ancestor of man, mouse and rat, motivated our investigations in conservation of common intervals. The data, based on the conservation of *synteny blocks*, yields permutations on 16 integers for each species.

This scenario, displayed in Fig. 2, has a remarkable feature: there is a common interval between the three species, $\{5, 6\}$, that is not conserved in the intermediate permutations!

Human	1	2	3	4	$\boxed{5}$	$\boxed{6}$	7	8	9	10	11	12	13	14	15	16
	1	2	3	4	$\boxed{5}$	$\boxed{6}$	7	8	9	10	11	12	13	14	$\overline{15}$	16
	1	$\overline{3}$	$\overline{2}$	4	$\boxed{5}$	$\boxed{6}$	7	8	9	10	11	12	13	14	$\overline{15}$	16
	1	$\overline{3}$	$\overline{2}$	$\boxed{\overline{5}}$	$\overline{4}$	$\boxed{6}$	7	8	9	10	11	12	13	14	$\overline{15}$	16
	1	$\overline{3}$	$\overline{2}$	$\boxed{\overline{5}}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	$\overline{9}$	$\overline{8}$	$\overline{7}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16
Median	1	$\overline{3}$	9	10	11	12	$\boxed{5}$	2	$\overline{8}$	$\overline{7}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16
Mouse	$\boxed{\overline{5}}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16	1	$\overline{3}$	9	$\overline{10}$	11	12	$\overline{7}$	8	$\overline{2}$
	$\boxed{\overline{5}}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16	1	$\overline{3}$	9	10	11	12	$\overline{7}$	8	$\overline{2}$
	$\boxed{\overline{5}}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16	1	$\overline{3}$	9	10	11	12	7	8	$\overline{2}$
	$\boxed{\overline{5}}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	$\overline{9}$	3	$\overline{1}$	$\overline{16}$	15	$\overline{14}$	$\overline{13}$	$\overline{4}$	$\boxed{6}$	7	8	$\overline{2}$
	1	$\overline{3}$	9	10	11	12	$\boxed{5}$	$\overline{16}$	15	$\overline{14}$	$\overline{13}$	$\overline{4}$	$\boxed{6}$	7	8	$\overline{2}$
Median	1	$\overline{3}$	9	10	11	12	$\boxed{5}$	2	$\overline{8}$	$\overline{7}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16
Rat	$\overline{13}$	$\overline{4}$	$\boxed{5}$	$\boxed{\overline{6}}$	$\overline{12}$	$\overline{8}$	$\overline{7}$	2	1	$\overline{3}$	9	10	11	14	$\overline{15}$	16
	$\overline{13}$	$\overline{4}$	$\boxed{6}$	$\boxed{\overline{5}}$	$\overline{12}$	$\overline{8}$	$\overline{7}$	2	1	$\overline{3}$	9	10	11	14	$\overline{15}$	16
	$\overline{13}$	$\overline{4}$	$\boxed{6}$	7	8	12	$\boxed{5}$	2	1	$\overline{3}$	9	10	11	14	$\overline{15}$	16
	$\overline{13}$	$\overline{4}$	$\boxed{6}$	7	8	$\overline{2}$	$\boxed{\overline{5}}$	$\overline{12}$	1	$\overline{3}$	9	10	11	14	$\overline{15}$	16
	$\overline{13}$	$\overline{4}$	$\boxed{6}$	7	8	$\overline{2}$	$\boxed{\overline{5}}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	$\overline{9}$	3	$\overline{1}$	14	$\overline{15}$	16
Median	1	$\overline{3}$	9	10	11	12	$\boxed{5}$	2	$\overline{8}$	$\overline{7}$	$\boxed{\overline{6}}$	4	13	14	$\overline{15}$	16

Fig. 2. The evolution scenario for human, mouse and rat X Chromosome of [10]

The first column of Table 2 gives the scores for the three corresponding pairwise scenarios, which are very low, even as low as 2/10 in the scenario transforming the rat chromosome X into the human chromosome X. The loss of the common interval $\{5, 6\}$ in the intermediate species clearly has a damaging effect on the scores. It also induces three independent reconstructions of this interval along the three branches that go from the median ancestor to the human, to the mouse, and to the rat.

In such a situation, it is possible to question several hypothesis of the model: the parsimony assumption, the evolutionary model, that considers only inversions, or, more simply, the data. Questioning the data was the easiest experiment, since the positions of the blocks were available. We soon realized that the mouse assembly used to construct the data, (assembly 30 of UCSD), differed from a more recent version, (assembly 32 of UCSD) notably on the respective position of synteny blocks 5 and 6, that are transposed[3].

Using the same synteny blocks in the order given by assembly 32 of mouse, and MGR we obtained an alternate scenario, with much better scores, as the second column of Table 2 shows: the total score increased from 10/30 to 18/30. Finally, we defined our own data set, that resulted in three permutations on 22 elements, as follows: we considered the genes of chromosome X in human (assembly 35.1 of NCBI), mouse (assembly 33.1 of NCBI) and rat (assembly 2.1 of NCBI), and we identified the genes common to the three genomes on the basis of their functional annotation, by using both confirmed and predicted annotations. The scenario we computed on this data set, described in Appendix A, is displayed in Figure 3, and the corresponding scores in the third column of Table 2

Human	14	$\overline{12}$	13	$\overline{15}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	5	$\overline{6}$	$\overline{16}$	$\overline{11}$	$\overline{10}$	17	$\overline{18}$	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	$\overline{21}$	22
	14	$\overline{12}$	13	$\overline{15}$	2	3	4	5	$\overline{6}$	$\overline{16}$	$\overline{11}$	$\overline{10}$	17	$\overline{18}$	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	$\overline{21}$	22
	14	$\overline{12}$	13	$\overline{15}$	2	3	4	5	$\overline{6}$	10	11	16	17	$\overline{18}$	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	$\overline{21}$	22
	14	12	13	$\overline{15}$	2	3	4	5	$\overline{6}$	10	11	16	17	$\overline{18}$	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	$\overline{21}$	22
	$\overline{14}$	12	13	$\overline{15}$	2	3	4	5	$\overline{6}$	10	11	16	17	$\overline{18}$	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	$\overline{21}$	22
	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{15}$	2	3	4	5	$\overline{6}$	10	11	16	17	18	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	$\overline{21}$	22
	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{15}$	2	3	4	5	$\overline{6}$	10	11	16	17	18	19	$\overline{9}$	8	$\overline{7}$	$\overline{1}$	20	21	22
	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{15}$	7	$\overline{8}$	9	$\overline{19}$	$\overline{18}$	$\overline{17}$	$\overline{16}$	$\overline{11}$	$\overline{10}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{15}$	10	11	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
Median	10	11	12	13	14	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22

Mouse	13	10	$\overline{12}$	$\overline{11}$	$\overline{8}$	9	$\overline{19}$	$\overline{18}$	$\overline{17}$	$\overline{16}$	$\overline{15}$	$\overline{14}$	$\overline{7}$	$\overline{22}$	$\overline{21}$	$\overline{20}$	1	2	$\overline{3}$	$\overline{4}$	5	$\overline{6}$
	13	10	$\overline{12}$	$\overline{11}$	$\overline{8}$	9	$\overline{19}$	$\overline{18}$	$\overline{17}$	$\overline{16}$	$\overline{15}$	$\overline{14}$	$\overline{7}$	$\overline{22}$	$\overline{21}$	$\overline{20}$	1	2	3	$\overline{4}$	5	$\overline{6}$
	13	10	$\overline{12}$	$\overline{11}$	$\overline{8}$	9	$\overline{19}$	$\overline{18}$	$\overline{17}$	$\overline{16}$	$\overline{15}$	$\overline{14}$	$\overline{7}$	$\overline{22}$	$\overline{21}$	$\overline{20}$	1	2	3	4	5	$\overline{6}$
	13	10	$\overline{12}$	$\overline{11}$	$\overline{8}$	9	$\overline{19}$	$\overline{18}$	$\overline{17}$	$\overline{16}$	$\overline{15}$	$\overline{14}$	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
	13	10	$\overline{12}$	$\overline{11}$	14	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
	$\overline{10}$	$\overline{13}$	$\overline{12}$	$\overline{11}$	14	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
	$\overline{10}$	11	12	13	14	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
Median	10	11	12	13	14	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22

Rat	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	1	2	3	4	5	6	7	$\overline{8}$	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	1	2	3	4	5	$\overline{6}$	7	$\overline{8}$	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	$\overline{19}$	$\overline{18}$	$\overline{17}$	$\overline{16}$	$\overline{15}$	$\overline{14}$	$\overline{13}$	$\overline{12}$	$\overline{11}$	$\overline{10}$	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22
Median	10	11	12	13	14	15	16	17	18	19	$\overline{9}$	8	$\overline{7}$	6	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	20	21	22

Fig. 3. The scenario from our 22 blocks data set

The conservation score we proposed and illustrated in this section is a first attempt to measure the conservation of combinatorial structure in evolution

[3] The mouse assembly 33 of UCSD agrees with the positions given in assembly 32.

scenarios, but raises many interesting questions. For example, the possibility to weight inversions that break common interval with respect to the position of the corresponding edge in the tree, or with respect to the number of broken intervals, should be considered. The interpretation of scores on permutations of different sizes, as we have in our example, is another interesting question.

	Mouse 30	Mouse 32	Mouse 32 + 22 blocks
Human to Mouse	4/10	8/10	15/18
Human to Rat	2/10	6/10	13/15
Rat to Mouse	4/10	4/10	5/11
Total	10/30	18/30	33/44

Table 2. Conservation scores of the scenario presented in [10], of a new scenario using the assembly 32 of the mouse, and of the scenario based on our gene blocks.

5 Perfect Scenarios

A *perfect scenario* is a scenario in which no inversion breaks a common interval. The construction of perfect scenarios is discussed in [9], where the problem is shown to be computationally difficult. Perfect scenarios always exist between two permutations, but are not necessarily optimal. They can even be trivial when the two permutations have few common intervals. For example, any scenario between permutations (1 2 3 4) and (3 1 4 2) is perfect. However, permutations that arise from genomic data of relatively close species share lots of common intervals, and some rearrangement scenarios have striking features in terms of structure conservation.

In this section, we study a class of perfect scenarios, called *commuting scenarios*, and we show that deciding the existence of optimal commuting scenario, and constructing them, can be done in linear time.

5.1 Commuting scenarios

Definition 4. *Let $r_1, \ldots, r_k$ be a sequence of inversions that transforms a permutation P into a permutation Q. The sequence $r_1, \ldots, r_k$ is a* commuting scenario *if, for every $i, j \in [1..k]$, the inversions r_i and r_j commute and are distinct.*

A beautiful example of a commuting scenario, is given in [7] where a region on human chromosome 17 (denoted by H below) is compared to a region on mouse chromosome 11 (denoted by M). The resulting permutations are:

$$H = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 17\ 18\ 19),$$
$$M = (7\ \overline{8}\ 6\ \overline{5}\ 4\ \overline{3}\ 1\ 2\ \overline{10}\ \overline{9}\ 11\ 12\ \overline{13}\ 16\ \overline{15}\ 14\ 17\ \overline{18}\ 19),$$

The mouse chromosome M can be obtained from chromosome H by the commuting scenario of Fig. 4, in which all the inversions are identified by underlining the set of inverted integers. Note also that we choose to represent the scenario by inversions applied to the identity permutation. This will be helpful in proving and understanding properties of commuting scenarios.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Fig. 4. A commuting scenario transforming H into M.

The fact that one could simultaneously underline all inversions in Fig. 4 is a direct consequence of the fact that all inversions commute, and implies the following lemma. This example highlights many properties of commuting scenarios. For example, applying the inversion from the largest to the smallest transforms H into M by always inverting segments of H composed of consecutive integers. More important for us is the following lemma.

Lemma 1. *Let $\mathcal{S}$ be an optimal commuting scenario between two permutations P and Q. An interval I is a common interval of P and Q if and only if all inversions of $\mathcal{S}$ commute with I.*

Proof. First, it is immediate that if each inversion of $\mathcal{S}$ commute with I, then I is a common interval between P and Q.

On the other hand, since $\mathcal{S}$ is a commuting scenario, one can first apply all inversions that commute with I, which leads to a permutation P' in which I is an interval. Let R be the set of remaining inversions, those that do not commute with I. If all inversions in R are disjoint, there are at most two of them, and it is easy to see that applying them to P' yields a permutation in which I is no longer an interval.

Suppose R contains at least two non-disjoint intervals, and that these intervals intersect I at its right extremity – the argument is completely symmetrical if we consider the left extremity. Let r be the largest interval that intersect I at its right extremity, s the second largest, and I' the non-empty set of elements of I that are at the left of r. Since the scenario is optimal, s is strictly contained in r. Therefore, r can be partitioned into disjoint intervals, u, s and v, such that u is contained in I, v is disjoint from I, and at least one of u and v is non-empty.

Applying both r and s to P' exchange the intervals u and v, leaving s in the middle. By the choice of r and s, this structure will remain unchanged for the rest of the sorting procedure, except for possible inversions within s. Note also that all elements of I' will remain to the left of r. If u is not empty, the elements of s that are not in I will end up between I' and u. If v is not empty, the elements of v will end up between I' and $s \cap I$. Thus I is not an interval in Q, and cannot be a common interval of P and Q.

□

Proposition 2. *Optimal commuting scenarios between two permutations P and Q conserve all common intervals of P and Q in all intermediate permutations.*

Proof. This follows immediately from Lemma 1. □

The scenario of Fig. 4 is also an optimal scenario. It is not true, in general, that if an optimal commuting scenario exists, then all optimal scenarios are commuting. An example is again given by data from human and mouse. Consider the first 8 segments of H and M: one can transform H into M with the sequence of inversions of Fig. 5, that is an optimal scenario, but not a commuting one.

$$1\ 2\ 3\ 4\ 5\ 6\ 7\ 8$$

$$1\ 2\ \overline{7}\ 3\ \overline{4}\ 5\ \overline{6}\ 8$$

$$7\ \overline{2}\ \overline{1}\ 3\ \overline{4}\ 5\ \overline{6}\ 8$$

$$7\ \overline{8}\ 6\ \overline{5}\ 4\ \overline{3}\ 1\ 2$$

Fig. 5. A non commuting scenario transforming the first 8 segments of H into M.

Deciding whether an optimal commuting scenario exists is therefore not a trivial question. We give, in the following section, a characterization of permutations that admit optimal commuting scenarios.

5.2 Existence of Optimal Commuting Scenarios

The results of this section rely on many of the concepts that have been developed around the sorting by inversion problem. The terminology and conventions we follow are presented in [5].

Note. In this section we consider signed permutations of $\{0, 1, \ldots, n\}$ that start with 0 and end with n.

A first remark is that, since an optimal commuting scenario does not break any common interval by Proposition 2, such a scenario can only exist for permutations that can be optimally sorted component by component. We first settle the case of oriented components in Theorem 1 that relates three fundamental types of intervals: common intervals, inversions of a commuting scenario, and the elementary intervals of the sorting by inversion theory, that appear here as the vertices of the overlap graph.

Theorem 1. *Let C be an oriented component of a permutation. The three following statements are equivalent.*

1. *C can be sorted by an optimal commuting scenario.*
2. *The overlap graph of C is a tree.*
3. *C can be sorted by an optimal scenario in which each inversion is a common interval.*

Before proving Theorem 1, we establish some properties of overlap graphs.

Lemma 2. *Any leaf of an overlap graph is an oriented interval, and a common interval.*

Proof. A leaf is an interval that overlaps only one other interval. Let m and M be the minimal and maximal values of a non-empty interval I_p. Then I_p always overlaps both I_{m-1} and I_M. If I_p is a leaf, we must have either $p = M$ or $p = m - 1$, implying in both cases that the interval is oriented, since it contains exactly one of its extremities. If I_p is not a common interval, then $|I_p| \geq 2$ and I_p does not contain all the integers between m and M. Let m' and M' be, respectively, the smallest and largest missing integers. Note that one can have $m' = M'$. Then I_p overlaps $I_{m'-1}$ and $I_{M'}$, and therefore can not be a leaf. $\square$

Lemma 3. *Erasing a leaf of an overlap graph that is a tree is always a sorting inversion.*

Proof. Erasing a leaf never disconnects a tree, and we only have to check that the tree contains at least another oriented interval after the leaf has been erased. If a leaf I_q overlaps I_p, and I_p is unoriented, then applying I_q orients I_p. If I_p is oriented and is the only other leaf (i.e, the graph contains only two vertices I_p and I_q), then $I_p = I_q$, and the component is sorted after erasing I_q. If I_p is connected to exactly one other node, I_p would become an unoriented leaf in the new tree, which is impossible by Lemma 2. Thus I_p is connected to at least two other nodes, implying at least two other leaves, therefore two other oriented intervals. □

Proof (Theorem 1).

(1) $\Longrightarrow$ (2). Suppose that an oriented component can be sorted by an optimal commuting scenario. We will show, by induction on the length of the scenario, that it must be a tree. It is certainly true for scenarios of length 1, since the overlap graph has two nodes. We will show that there always exists an inversion r that does not contains any other inversion of the scenario, and that necessarily creates an adjacency. Since r can be applied first, this implies that r is an elementary interval and a leaf in the overlap graph, and the result follows by induction. If all inversions are disjoint, then the leftmost certainly creates an adjacency. Otherwise, let r be a smallest inversion, in terms of the number of elements it inverts, included in another inversion, and s the smallest inversion containing r. As the inclusion of r in s is strict, then r creates an adjacency between one of its elements and an element of s not included in r.

(2) $\Longrightarrow$ (3). It follows immediately from Lemmas 2 and 3: erasing a leave always yields a tree.

(3) $\Longrightarrow$ (1). Let $\mathcal{S}$ be a set of reversals that transforms permutation P into perrmutation Q, and such that all inversions of $\mathcal{S}$ are common intervals of P and Q. Apply to P a maximal subset R of commuting inversions from $\mathcal{S}$ yielding permutation P'. By Lemma 1, all common intervals of P and P' commute with all inversions of R. Let R' be the set of remaining inversions. If R' is not empty, there is at least one inversion s that is an interval of P', and that does not commute with an inversion of R. Therefore, s cannot be an interval of P, and is not a common interval of P and Q.

□

In order to have a characterization of all permutations that admits optimal commuting scenarios, we must next deal with unoriented components. In this case, one inversion is allowed to orient the component, but, as we saw in Theorem 1, the overlap graph of the resulting component must be a tree, which restricts severely the structure of the overlap graph of the original unoriented component: it must contain only one cycle. This imposes to unoriented components the following *reduced* form [4].

[4] A component is *reduced* if all the smaller component contained in it have been sorted, and all the resulting adjacencies collapsed into single elements [3].

Theorem 2. *An unoriented component admits an optimal commuting scenario if and only if it can be reduced to a permutation of the form*

$$(0 \;\; 2k \;\; 2k-1 \;\; \ldots \;\; 3 \;\; 2 \;\; 1 \;\; 2k+1).$$

Proof. Let P be a positive permutation P with n breakpoints and one component. If P can be optimally sorted with a commuting scenario $\mathcal{S}$, then each inversion of $\mathcal{S}$ can be applied first, and must create a permutation whose overlap graph is a tree. We will show that the inversions of S are either single elements, or all the elements of the interval $[1..n-1]$, implying that P is of the form $(0 \;\; n-1 \;\; n-2 \;\; \ldots 3 \;\; 2 \;\; 1 \;\; n)$, and that the length of a commuting scenario is n.

Let r be an inversion of $\mathcal{S}$, with minimum element m and maximum element M, then applying r to P creates the oriented elementary intervals I'_{m-1} and I'_M, that are leaves of the resulting overlap graph. By Lemmas 1 and 2, those intervals must commute with r, implying that r also commutes with I_{m-1} and I_M.

If r commutes with I_{m-1}, then either $m-1$ is immediately to the left of interval r, or m is the first element of r. Similarly, if r commutes with I_M, then either $M+1$ is immediately to the right of interval r, of M is the last element of r.

If m is the first element of r, and M the last, interval r will be a component unless $m = M$. This case produces the inversions consisting of a single element. If $m-1$ is immediately to the left of r, and $M+1$ is immediately to the right, then $[m-1..M+1]$ is a component, thus $m-1=0$, and $M+1=n$. This case produces the inversion of the interval $[1..n-1]$.

Finally, if $m-1$ is immediately to the left of r, and M the last element of r, then $[m-1..M]$ is a component, implying that $M = n$, which is impossible. Similarly, $M+1$ is immediately to the right of r, and m the first element of r, implies $m = 0$, which is also impossible.

Applying all the possible n commuting inversions to the identity permutation yields the permutation: $(0 \;\; n-1 \;\; n-2 \;\; \ldots 3 \;\; 2 \;\; 1 \;\; n)$. If n is odd, the inversion distance is n, thus the permutation can be sorted by an optimal commuting scenario. However, if n is even, the permutation has two cycles, and there exists an optimal scenario of length $n-1$. □

5.3 Algorithm

We now have all the elements needed to construct a linear time algorithm that will decide if a permutation P of size n can be sorted with an optimal commuting scenario, and that will compute the necessary information to obtain such a scenario, if it exists.

General overview. The overlap graph can be processed component by component, and Theorem 2 addresses the case of unoriented components.

In the case of an oriented component, whose overlap graph C has k vertices, the algorithm given in Fig. 6 decides, in linear time, if C can be sorted by a commuting scenario. If such a scenario exists, the algorithm computes, in linear time, the order in which the vertices of the overlap graph must be erased to produce a commuting set of inversions.

1. Build iteratively the edges of C as long as there are at most $k-1$.
2. If C has at least k edges, then C is not a tree.
3. Else, remove iteratively the leaves of C, which produces the sequence of inversions necessary to sort C.

Fig. 6. Algorithm 1 (Main algorithm).

As one can see, the core of this algorithm is step 1., that ensures that one can decide if the component can be sorted with a commuting scenario after considering at most k edges. The last step, that produces the scenario, can clearly be done in a single traversal of the overlap tree, and thus in $O(k)$ time. So we need to describe how we identify at most k edges of the overlap graph in time $O(k)$.

Computing edges of the overlap graph. Let (ℓ_i, r_i) be the indices, respectively, of the left point and right point of the elementary interval I_i of C. The first step is to compute a sequence S of the $2k$ ℓ_i's and r_i's in such a way that the following property holds: two intervals I_p and I_q overlap if and only if in S ℓ_q appears between ℓ_p and r_p and r_q appears after r_p. This can be done in $\Theta(k)$ worst-case time.

Once the sequence S is built, the following algorithm computes at most m edges of the overlap graph during a single pass on S. One denotes by S_i the i^{th} element of S and m the maximum number of edges one wants to produce.

1. Let $i = 1$.
   // Invariant: there are only ℓ_q's at the left of S_i, and all of them have their corresponding r_q's at the right of S_i or at S_i.
2. While $i \leq 2n$ and less than m edges have been computed do
   3. If $S_i = r_p$ for some p then
      4. Let $S_j = \ell_p$.
      5. For every ℓ_q located between S_j and S_i do add an edge (p, q).
      6. Remove from S the elements ℓ_p and r_p.
         // This last step ensures the invariant still holds

Fig. 7. Algorithm 2 (Computing edges of the overlap graph).

Using the appropriate data structures to encode S, such as a double-linked list, one can implement this algorithm in order that instruction 6, that removes elements of S, is done in $\Theta(1)$ time, and that instruction 5, that visits all elements between ℓ_p and r_p, is done in time proportional to the number of these elements. The invariant ensures that the elements visited between ℓ_p and r_p are exactly the ℓ_q's such that the corresponding r_q's are located after r_p. This leads to the following lemma:

Lemma 4. *For every m, Algorithm 2 computes at most m edges of C in $\Theta(m+k)$ worst-case time.*

Theorem 3. *It can be decided, in $\Theta(n)$ time and space, whether a signed permutation P on n elements can be sorted by an optimal commuting scenario. If an optimal commuting scenario exists, one can compute the corresponding sequence of oriented inversions in $\Theta(n)$ time and space.*

Proof. Given P, we can process it component by component. The case of unoriented components, that can be detected in $\Theta(n)$ time [4], has been addressed in Theorem 2 and can be solved in $\Theta(n)$ time. Next, one needs to compute the set of vertices of each component of the overlap graph, and this can be done in linear time using [4, 1]. Then we apply Algorithm 1, where step 1 is done with Algorithm 2, and step 3, if necessary, is done during a depth-first traversal of the overlap graph of C where each leaf is processed – the corresponding inversion is added to the scenario – during its first visit. Steps 1 and 3 take $\Theta(k)$ time if the current component has k vertices, by definition of a depth-first traversal and Lemma 4, which leads to a total $\Theta(n)$ time complexity to build the forest of trees that composes the overlap graph and iteratively remove the leaves of this forest. □

6 Conclusion

We described in this paper a class of perfect scenarios, the commuting scenarios, and we showed that one can decide in linear time whether a signed permutation can be sorted by an optimal commuting scenario. However since a perfect scenario is not necessarily commuting, it is still an open question to decide in polynomial time if a permutation can be sorted by an optimal perfect scenario. It would also be interesting to have more information on how large is the class of permutations that can be sorted by commuting scenarios. This would help in assessing the significance of optimal commuting scenarios with respect to other optimal scenarios. Here, we focused on optimal commuting scenarios, and the class of non-optimal commuting scenarios should be investigated. Indeed, every permutation can be sorted by a perfect scenario, but it is not true that every permutation can be sorted by a commuting scenario. Finally, it should also be noted that the best time complexity for computing an optimal scenario for a general permutation is currently $(n\sqrt{n\log(n)})$ [16]. Our algorithm is the first, as far as we know, that achieves linear-time complexity for a non-trivial class of signed permutations.

References

1. D. A. Bader, B. M. E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *J. Comp. Biol.*, 8(5):483–491, 2001.

2. A. Bergeron. A very elementary presentation of the hannenhalli-pevzner theory. In *CPM 2001*, volume 2089 of *Lecture Notes in Comput. Sci.*, pages 106–117, 2001. (Extended version to appear in Discrete Applied Math.).
3. A. Bergeron, C Chauve, T. Hartman, and K. St-Onge. On the properties of sequences of reversals that sort a signed permutation. In *JOBIM 2002*, pages 99–108, 2002.
4. A. Bergeron, S. Heber, and J. Stoye. Common intervals and sorting by reversals: A marriage of necessity. *Bioinformatics*, 18(Suppl. 2):S54–S63, 2002. (ECCB 2002).
5. A. Bergeron, J. Mixtacki, and J. Stoye. Reversal distance without hurdles and fortresses. In *CPM 2004*, volume 3109 of *Lecture Notes in Comput. Sci.*, pages 388–399, 2004.
6. G. Bourque and P. A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res.*, 12(1):26–36, 2002.
7. G. Bourque, P. A. Pevzner, and G. Tesler. Reconstructing the genomic architecture of ancestral mammals: Lessons from human, mouse, and rat genomes. *Genome Res.*, 14(4):507–516, 2004.
8. A. Caprara. Formulations and hardness of multiple sorting by reversals. In *RECOMB 1999*, pages 84–94, 1999.
9. M. Figeac and J.-S. Varré. Sorting by reversals with common intervals. In *WABI 2004*, volume 3240 of *Lecture Notes in Bioinformatics*, pages 26–37, 2004.
10. R. A. Gibbs et al. Genome sequence of the brown norway rat yields insights into mammalian evolution. *Nature*, 428:493–521, 2004.
11. 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.
12. S. Heber and J. Stoye. Finding all common intervals of k permutations. In *CPM 2001*, volume 2089 of *Lecture Notes in Comput. Sci.*, pages 207–218, 2001.
13. H. Kaplan, R. Shamir, and R. E. Tarjan. A faster and simpler algorithm for sorting signed permutations by reversals. *SIAM J. Computing*, 29(3):880–892, 1999.
14. 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 *WABI 2002*, volume 2452 of *Lecture Notes in Comput. Sci.*, pages 521–536, 2002.
15. P. A. Pevzner and G. Tesler. Genome rearrangements in mammalian evolution: Lessons from human and mouse genomes. *Genome Res.*, 13(1):37–45, 2003.
16. E. Tannier and M.-F. Sagot. Sorting by reversals in subquadratic time. In *CPM 2004*, volume 3109 of *Lecture Notes in Comput. Sci.*, pages 1–13, 2004.
17. G. Tesler. GRIMM: genome rearrangements web server. *Bioinformatics*, 18(3):492–493, 2002.
18. T. Uno and M. Yagiura. Fast algorithms to enumerate all common intervals of two permutations. *Algorithmica*, 26(2):290–309, 2000.

Appendix A. The 22 gene blocks of the new human/mouse/rat scenario

We give here the list of the first gene of the 22 blocks of genes we used in our study of the human/mouse/rat X chromosome evolution. The below table has the following format: the first column gives the integers associated to the 22 genes we consider, the next three columns present the rat data (gene/locus name, orientation and position) followed by three columns for the mouse and three columns for the human (assembly 34.3 for the human genome, assembly 32.1 for the mouse genome and assembly 2.1 for the rat genome).

1	Birc4	-	3013140	Birc4	+	33413632	BIRC4	+	121691803
2	Syn1	+	12517799	Syn1	-	19413738	SYN1	-	46478245
3	Sytl5	-	24886677	Sytl5	-	8475502	SYTL5	+	36924044
4	Cybb	-	25568041	Cybb	-	7985479	CYBB	+	36670221
5	Ebp	+	26384668	Ebp	-	6745931	EBP	+	47426239
6	Gdf-9b	-	29057402	Bmp15	-	4973481	BMP15	+	49570590
7	Pls3	-	29923507	Pls3	-	65568249	PLS3	+	113559762
8	Il13ra2	-	30559317	Il13ra2	-	135823846	IL13RA2	-	113002791
9	Dcx	+	34649731	Dcx	-	132129216	DCX	-	109300978
10	Ragb	-	38406475	LOC245670	+	141432864	RRAGB	+	54711109
11	Pfkfb1	+	39892044	Pfkfb1	+	138292880	PFKFB1	-	53926381
12	LOC317435	+	42360000	APXL	+	140920000	APXL	-	9250000
13	Mid1	-	44805419	Mid1	+	157816113	MID1	-	9827653
14	LOC302711	+	63450000	Tbl1x	+	67700000	Tbl1x	+	8845000
15	Dmd	+	71574635	Dmd	+	73462214	DMD	-	30498771
16	Maged1	-	82127336	Maged1	-	84274362	MAGED1	+	50553552
17	Arhgef9	-	82667648	Arhgef9	-	84771512	ARHGEF9	-	61721639
18	Slc16a2	-	91773989	Slc16a2	-	93602023	SLC16A2	+	72507876
19	Atrx	-	93979545	Atrx	-	95705534	ATRX	-	75517065
20	Xpnpep2	+	134548393	Xpnpep2	+	39428571	XPNPEP2	+	127578549
21	LOC367956	+	147367900	Gm366	+	51313500	LDOC1	-	138963500
22	Fmr1	+	154829464	Fmr1	+	58282553	FMR1	+	145661196

This data set induces the three following permutations (repectively, from top to bottom, for the rat, the mouse and the human), where the mouse chromosome X is represented reversed, in order to correspond to the evolution scenario given in Appendix C:

$$\begin{array}{rcl}
\text{Rat} = & (& 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 16\ 18\ 19\ 20\ 21\ 22\ ) \\
\text{Mouse} = & (& 13\ 10\ \overline{12}\ \overline{11}\ \overline{8}\ 9\ \overline{19}\ \overline{18}\ \overline{17}\ \overline{16}\ \overline{15}\ \overline{14}\ \overline{7}\ \overline{22}\ \overline{21}\ \overline{20}\ 1\ 2\ \overline{3}\ \overline{4}\ 5\ \overline{6}\ ) \\
\text{Human} = & (& 14\ \overline{12}\ 13\ \overline{15}\ \overline{4}\ \overline{3}\ \overline{2}\ 5\ \overline{6}\ \overline{16}\ \overline{11}\ \overline{10}\ 17\ \overline{18}\ 19\ \overline{9}\ 8\ \overline{7}\ \overline{1}\ 20\ \overline{21}\ 22\ )
\end{array}$$