

Ultra-Perfect Sorting Scenarios

Aïda Ouangraoua¹, Anne Bergeron², and Krister M. Swenson^{2,3}

¹ INRIA LNE, LIFL, Université Lille 1, Villeneuve d'Ascq, France

² Lacim, Université du Québec à Montréal, Montréal, Canada

³ Department of Mathematics and Statistics, University of Ottawa, Canada

Abstract. Perfection has been used as a criteria to select rearrangement scenarios since 2004. However, there is a fundamental bias towards extant species in the original definition: ancestral species are not bound to perfection. Here we develop a new theory of perfection that takes an egalitarian view of species, and apply it to the complex evolution of mammal chromosome X.

1 Introduction

In mathematical biology, the genome sorting problem is to find a sequence of rearrangement operations that transforms one genome into another. The type of rearrangement operations is fixed, and a sorting sequence of operations is called a *scenario*. Given two genomes, there can be an exponential number of scenarios, which makes difficult the choice of one particular scenario, even among those of minimum length. Parsimony is only one of the criteria that can be used for the selection of a scenario, and there are many alternatives that are worth exploring.

Here we consider the problem of *perfect sorting* which was initially stated roughly as follows: given two genomes, find a sorting scenario between the genomes that preserves common genomic segments in intermediate states of the transformation. Such scenarios are called *perfect* and this problem was first introduced under the inversion rearrangement model by [7] who showed the NP-hardness of the problem. It was later shown that for some classes of instances, the problem could be solved in polynomial time [1,2,6,12]. More recently [3], the problem was explored under the double-cut-and-join (DCJ) rearrangement model, using a less stringent definition of perfection that allows temporary circular chromosomes.

In this paper, we address the problem of perfect sorting by DCJ under the initial definition of perfection. We also reexamine the original idea of perfection which applies only to the two compared extant species. What about all intermediate species that are generated by the scenario? Any good biological argument in favor of perfect scenarios would apply to any pair of these intermediate species. In this paper, we intend to correct this injustice.

We introduce a new, more restrictive class of perfection, called *ultra-perfection* with the corresponding problem: given two genomes, find a sorting scenario such that any sub-scenario is perfect. Our main results are the description of combinatorial properties of ultra-perfection that leads to a polynomial time algorithm

for computing ultra-perfect scenarios between genomes, and the construction of a *near ultra-perfect* scenario between human, mouse and rat chromosome X.

The paper is organized as follows: in Section 2, we give the definitions of genomes, common intervals, rearrangement scenarios and ultra-perfection. In Section 3, we characterize ultra-perfection in terms of commutation of inversion scenarios, which leads to an algorithm for computing ultra-perfect scenarios. In section 4, we study the ultra-perfection of scenarios between the human and rodent chromosomes X. We conclude in Section 5 with remarks on the scenario described for human and rodents chromosome X, and on the algorithmic complexity of finding ultra-perfect or near ultra-perfect scenarios between several species.

2 Models and Definitions

In this section, we give the main definitions and notation that are used in the paper: genomes, inversions, double-cut-and-join operations, commuting inversions, perfect and ultra-perfect scenarios.

2.1 Genomes

Genomes are compared by identifying homologous segments along their DNA sequences, called *blocks*, organized in circular or linear chromosomes. A genome is *circular* (resp. *linear*) if it is only composed of circular (resp. linear) chromosomes. Each genome contains exactly one occurrence of each block, and the order and orientation of the blocks may differ between genomes. A linear chromosome will be represented by an ordered sequence of signed integers, one for each block, flanked by the unsigned block \circ at each end, and a circular chromosome will be represented by a circularly ordered sequence of signed integers. For example, genome $(-5 \ 7 \ 1 \ -3 \ 2) (\circ \ -6 \ 4 \ 8 \ 9 \circ)$ consists of one circular chromosome and one linear chromosome.

An *adjacency* in a genome is a pair of consecutive blocks. Since a chromosome can be read in two directions; the adjacencies $(x \ y)$ and $(-y \ -x)$ are equivalent. Moreover, since the block \circ is unsigned, the adjacencies $(\circ \ y)$ and $(-y \ \circ)$ are equivalent. An *interval* in a genome is a set of blocks that appear consecutively in the genome. A *common interval* between genomes A and B is an interval that exists in both A and B . A *maximal common interval* between A and B is a common interval that is not included in any other common interval between A and B .

2.2 Rearrangement Scenarios

In this paper, we consider two models of rearrangements: the inversion model and the double-cut-and-join model. An *inversion* of a set of contiguous blocks reverses the order of those blocks and change their signs. A *double-cut-and-join* (DCJ) operation on a genome A cuts two different adjacencies in A and glues

pairs of the four exposed extremities to form two new adjacencies, no other adjacency is altered. The *circularization* of a linear chromosome $(\circ \dots x \dots y \circ)$ is the DCJ operation that cuts adjacencies $(\circ x)$ and $(y \circ)$ to produce $(y x)$ and $(\circ \circ)$, thus creating the circular chromosome $(x \dots y)$. The opposite operation called a *linearization* is a DCJ operation that transforms a circular chromosome into a linear chromosome.

For example, a DCJ operation on genome $(-5 \ 7 \ 1 \ -3 \ 2) (\circ \ -6 \ 4 \ 8 \ 9 \circ)$ that cuts the adjacencies $(2 \ -5)$ and $(-6 \ 4)$ to form $(2 \ 4)$ and $(-6 \ -5)$ would produce genome $(\circ \ -6 \ -5 \ 7 \ 1 \ -3 \ 2 \ 4 \ 8 \ 9 \circ)$. Note that we consider the empty chromosome $(\circ \circ)$ to belong to any genome so that the DCJ on the circular genome $(-5 \ 7 \ 1 \ -3 \ 2)$ which cuts adjacencies $(7 \ 1)$ and $(\circ \circ)$ will produce the linear genome $(\circ \ 1 \ -3 \ 2 \ -5 \ 7 \circ)$.

Let A and B be two genomes. A *scenario* from A to B is a sequence of rearrangements that transforms A into B . An *inversion scenario* from A to B contains only inversions, and *DCJ scenario* contains only DCJ operations. Since an inversion can always be realized by one DCJ operation, an inversion scenario is always a DCJ scenario.

Note that the application of an inversion on a given genome only affects a single chromosome of the genome. Thus, an inversion scenario on a genome A can always be decomposed into a set of independent inversion scenarios, each acting on a single chromosome of genome A . In the following, we will restrict our study of inversion scenarios to unichromosomal genomes.

2.3 Perfection and Commutation

The following definition of perfection is used in [1,2,7,12]:

Definition 1 (Perfection). *A rearrangement scenario from genome A to genome B is perfect if all common intervals of A and B are also intervals in the intermediate genomes of the scenario.*

For the purposes of this paper we introduce a new, more restrictive class of perfection that we call *ultra-perfection*.

Definition 2 (Ultra-Perfection). *A rearrangement scenario from genome A to genome B is ultra-perfect if all sub-scenarios are perfect.*

The difference between the two notions is illustrated by the following example. Consider the two genomes $(\circ \ -3 \ -1 \ 4 \ 2 \circ)$ and $(\circ \ 1 \ 2 \ 3 \ 4 \circ)$ that have no common intervals, except trivial ones. Definition 1 implies that any inversion scenario between the two genomes is perfect. However, as we will show in Section 3, none of these scenario is ultra-perfect. In particular, the following scenario creates the common interval $\{2, 3, 4\}$ between chromosomes C_2 and C_4 , but destroys it in C_3 :

$$\begin{aligned} C_1 &= (\circ \ -3 \ \underline{-1 \ 4 \ 2} \circ) \\ C_2 &= (\circ \ -3 \ \underline{-2 \ -4 \ 1} \circ) \\ C_3 &= (\circ \ \underline{-3 \ -2 \ -1} \ 4 \circ) \\ C_4 &= (\circ \ 1 \ 2 \ 3 \ 4 \circ) \end{aligned}$$

As we will see, the notion of ultra-perfection is intimately related with commutation of inversions. We first recall the definition of commutation used in [1]: two sets of blocks commute if they are either disjoint or one is included in the other. Two inversions *commute* if their associated sets of blocks commute. An inversion scenario is *commuting* if all pairs of inversions contained in the scenario commute.

When considering circular chromosome, the definition of commutation must be adapted. Indeed, in a circular chromosome containing the set of blocks G , an inversion associated to a set $S \subset G$ produces the same result as an inversion associated to the set $G \setminus S$. We thus introduce the *circularly commuting* property defined as: given a set G of blocks, two subsets S and T of G *commute circularly* if S and T commute, or $G \setminus S$ and T commute.

Note that in [3] a less stringent definition of perfection is used for DCJ scenarios. In this version, the notion of common interval is relaxed to allow subsets of common intervals to form circular chromosomes.

3 Ultra-Perfect Scenarios

We now consider the problem of computing an ultra-perfect scenario between two genomes. In Section 3.1, we show that each maximal common interval can be considered independently. This implies an ultra-perfect scenario that first sorts each substring associated with a maximal common interval, and then sorts the whole genome in its final configuration. Section 3.2 then describes conditions for the existence of ultra-perfect DCJ scenarios between substrings associated to each maximal common interval.

Throughout the section we refer to a substring of a genome associated with an interval as a *segment* of the genome. The segment of a genome A induced by an interval I is denoted A_I .

3.1 Independent DCJ Sorting of Maximal Common Intervals

The following proposition states that, in any ultra-perfect DCJ scenario between genomes A and B , the segments induced by maximal common intervals of A and B are sorted independently.

Proposition 1. *Let I be a maximal common interval between genomes A and B . If \mathcal{S} is an ultra-perfect DCJ scenario from A to B , then there exists an equal length ultra-perfect scenario $\mathcal{S}' = \mathcal{S}_I T$ such that \mathcal{S}_I is an ultra-perfect DCJ scenario transforming A_I into B_I .*

Proof. Let C be a genome obtained by applying some (possibly empty) prefix of \mathcal{S} , and D be the genome obtained by applying some subsequent operations of \mathcal{S} on C . Since \mathcal{S} is ultra-perfect, I is an interval of C and D . Say that the operation transforming the intermediate genome C into the intermediate genome D *modifies* I if $C_I \neq D_I$.

Let f be an operation of \mathcal{S} such that f modifies I and the operation e preceding f does not modify I . If f cuts no adjacency created by e , then simply switch the order of e and f . Otherwise we replace e and f by two DCJs e' and f' such that e' precedes f' , through the following process: say e cuts the adjacency $(u v)$ and $(y x)$, u being a block belonging to I , to create $(u x)$ and $(y v)$, and that f then cuts $(u x)$ and $(t s)$ to create $(u s)$ and $(t x)$. Then take DCJs e' and f' such that e' cuts the adjacency $(u v)$ and $(t s)$ to create $(u s)$ and $(t v)$, and f' cuts $(t v)$ and $(y x)$ to create $(t x)$ and $(y v)$. In this way, any operations modifying I can be moved to the beginning of the DCJ scenario. Each move does not effect the ultra-perfection of the scenario since e' cannot create an interval that is later broken: any new interval created by e' (and later broken by some DCJ g) would have to include some elements of I and some adjacent elements to I . But this implies the existence of a larger interval that would be broken by g in the original scenario.

Moreover, at the end of this process, the scenario obtained can be decomposed into two sequences \mathcal{S}_I and \mathcal{T} such that all operations in \mathcal{S}_I modify I but no operation in \mathcal{T} modifies I . Then, \mathcal{S}_I is an ultra-perfect DCJ scenario between the segments A_I and B_I . \square

3.2 Ultra-Perfect Scenarios between Unichromosomal Genomes

Proposition 1 implies that, for each maximal common interval I of A and B , the ultra-perfect sorting of A_I into B_I can be examined independently from the rest of the scenario. We now give characterizations of the existence of ultra-perfect DCJ scenarios between unichromosomal genomes A and B by establishing properties of commuting inversion scenarios.

Ultra-perfect inversion scenarios. First we characterize ultra-perfect inversion scenarios.

Proposition 2. *An inversion scenario between two linear genomes is ultra-perfect if and only if the scenario is commuting.*

Proof. [2] shows that a commuting inversion scenario is always perfect. So, if a scenario \mathcal{S} is commuting, then all sub-scenarios of \mathcal{S} are commuting and thus perfect. Commutation then implies ultra-perfection.

Next, let \mathcal{S} be an ultra-perfect scenario. Suppose that \mathcal{S} is not commuting. Then there exists two inversions in \mathcal{S} , e preceding f , such that e and f do not commute but e and f commute with all inversions in \mathcal{S} between e and f .

Say, without loss of generality, that the genome looks like $UVWXY$ before applying inversion e where U , V , W , X , and Y represent sets of blocks. In this configuration $e = \{V, W\}$ and $f = \{V, X\}$, and all inversions in the scenario between e and f , then, do not change the relative order of V, W , and X to each other (since they commute with e and f). So e creates the order WVX while f creates the order WXV . But this contradicts the hypothesis that \mathcal{S} is ultra-perfect since the interval $\{W, X\}$ is destroyed by e and recreated by f . \square

In [2], it was shown that the commuting inversion scenarios between linear genomes could be characterized in terms of the structure of a tree, called the *strong interval tree*, representing the set of all common intervals. The *strong interval tree* of two linear genomes is a tree whose vertices are the common intervals that commute with any other common interval and there is an edge (I, J) between two vertices if $J \subset I$ and there exist no third vertex K such that $J \subset K \subset I$. The strong interval tree was first described in [8,11,4] and inspired from a data structure called the *PQ-tree*. PQ-trees are used to represent all consecutive-ones orderings of the columns of a matrix that has the consecutive-ones property.

In the remainder of the section we develop the counterpart characterization for DCJ scenarios by relating ultra-perfect DCJ scenarios to a tree representing the set of all common intervals between two circular genomes. We present the *circular common interval tree* that is the circular analogue of the strong interval tree. Circular common interval trees are inspired from an analogous structure of a PQ-tree, called a *PC-tree*. The PC-tree was introduced in [9,10] where it was used to represent all circular-ones orderings of the columns of a matrix that has the circular-ones property.

Circular common interval tree. Here, we define a tree representing the set of all common intervals between two circular genomes. Let G be a set of n blocks, and A and B be two circular genomes on G . A *circular common interval* of A and B is either a singleton block, or a subset I of G such that I is a common interval between A and B , and $|I| \neq n - 1$. The *circular strong intervals* of A and B are the circular common intervals of A and B that commute circularly with any other circular common interval.

The *circular common interval tree* of two circular genomes is then defined as follows:

Definition 3. *The circular common interval tree of two circular genomes A and B , denoted by $T(A, B)$ is defined as follows: the vertices of $T(A, B)$ are the circular strong intervals of A and B that commute with any other circular strong interval; a vertex J is a child of a vertex I if $J \subset I$, and there exist no third vertex K such that $J \subset K \subset I$.*

For example, let us consider the following circular genomes $A = (2\ 4\ 3\ 1\ 5)$ and $B = (1\ 2\ 3\ 4\ 5)$. The circular common intervals of A and B are $\{1, 5\}$, $\{1, 5, 2\}$, $\{4, 3\}$, $\{2, 4, 3\}$, and the singletons $\{1\} \dots \{5\}$. The circular strong intervals are all the circular common intervals. The vertices of the circular common interval tree $T(A, B)$ are $\{1, 5\}$, $\{4, 3\}$ and the singletons. $T(A, B)$ is depicted in Figure 1.a.

Given a vertex I of $T(A, B)$, two orderings, either both circular, or both linear, of the set of children of I can be inferred from the orderings in A and B of the set of blocks composing I . Following the notation of [2], the vertex I is *linear* if both orderings are identical or reciprocal, otherwise the vertex I is *prime*. If I is a linear vertex, I has a *positive sign* if both orderings are identical, and a *negative sign* if they are reciprocal (see Figure 1 for an example).

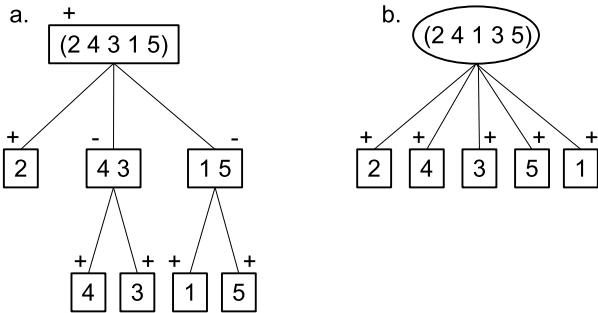


Fig. 1. The circular common interval trees of a. A and C , and b. B and C where genomes A, B, C are $A = (2 \ 4 \ 3 \ 1 \ 5)$, $B = (2 \ 4 \ 1 \ 3 \ 5)$, and $C = (1 \ 2 \ 3 \ 4 \ 5)$. Rectangular nodes correspond to linear vertices and round nodes to prime vertices. The signs of linear vertices are indicated by a + or - symbol.

A circular common interval tree is *definite* if its vertices are linear. For example, in Figure 1, the left-hand circular common interval tree is definite, while the right-hand one is not definite.

The definition of linear or prime vertices, and definite trees, also hold for the strong interval tree of two linear genomes. Given a vertex I of the strong interval tree of two linear genomes A and B , if I is linear (resp. prime), we also say that the common interval I between A and B is linear (resp. prime).

Ultra-perfect DCJ scenarios. In [2], it was shown that the existence of a perfect inversion scenario between unichromosomal linear genomes A and B was conditioned on properties of the strong interval tree of A and B : there exists a commuting scenario if and only if the strong interval tree is definite. In the following, we give the equivalent theorem for DCJ scenarios with circular common interval trees: there exists an ultra-perfect DCJ scenario if and only if the circular common interval tree is definite. We start by stating an obvious but useful property of ultra-perfect DCJ scenarios.

Property 1. Let A and B be unichromosomal genomes on the same set of blocks G . G is a common interval between A and B . So, if a DCJ scenario \mathcal{S} from A to B is ultra-perfect, then any operation in \mathcal{S} is either an inversion, or a circularization, or a linearization.

The *circular version* of a unichromosomal genome A is A itself if A is already a circular genome, otherwise it is the circular genome obtained by applying the circularization DCJ on A . We denote by A_c the circular version of a genome A .

Theorem 1. Let A and B be two unichromosomal genomes with the same set of blocks. There exists an ultra-perfect DCJ scenario from A to B if and only if the circular common interval tree of A_c and B_c is definite.

Proof. Let $T = T(A_c, B_c)$ be a circular common interval tree of A_c and B_c .

First, if T is definite, then an inversion scenario \mathcal{S} composed of inversions whose associated sets are the vertices of T that have a sign different from their parents is a commuting inversion scenario from A_c to B_c . So, \mathcal{S} is an ultra-perfect scenario. Say $e \mathcal{S} f$ is the DCJ scenario from A to B where e is the eventual circularization transforming A into A_c and f is the eventual linearization transforming B_c into B . It is easy to see that $e \mathcal{S} f$ is an ultra-perfect scenario from A to B .

Now, if T is not definite, let I be a vertex of T that is prime. Since a DCJ scenario from A to B contains only inversions, circularization and linearization, then there exists no ultra-perfect DCJ scenario that sorts A_I into B_I . \square

For example, consider the linear genomes $A = (\circ 3 \ 1 \ 5 \ 2 \ 4 \circ)$ and $B = (\circ 1 \ 2 \ 3 \ 4 \ 5 \circ)$. The circular common interval tree of the circular versions of A and B is depicted in Figure 1. Since it is a definite tree, then there exists the following ultra-perfect DCJ scenario:

$(\circ 3 \ 1 \ 5 \ 2 \ 4 \circ)$	<i>circularization</i>	$(4 \ -5 \ -1 \ 2 \ 3)$	<i>inversion of {1}</i>
$(3 \ 1 \ 5 \ 2 \ 4)$	<i>inversion of {4,3}</i>	$(4 \ -5 \ 1 \ 2 \ 3)$	<i>inversion of {5}</i>
$(-4 \ 1 \ 5 \ 2 \ -3)$	<i>inversion of {1,5}</i>	$(4 \ 5 \ 1 \ 2 \ 3)$	<i>linearization</i>
$(-4 \ -5 \ -1 \ 2 \ -3)$	<i>inversion of {4}</i>	$(\circ 1 \ 2 \ 3 \ 4 \ 5 \circ)$	
$(4 \ -5 \ -1 \ 2 \ -3)$	<i>inversion of {3}</i>		

4 Ultra-Perfection Meets Reality

The study of perfect rearrangement scenarios was, in large part, motivated by scenarios of inversions linking the human, mouse and rat chromosomes X [1]. In that first study, the presence of prime intervals arising from the comparison between human and rodents was largely ignored, since any inversion scenario that sorts a prime interval is perfect. In the framework of ultra-perfection, it makes sense to revisit this fundamental example.

In this section, we will show that, although no ultra-perfect scenario exists between human and rodents, there exist a scenario which is minimally disruptive. We will first describe a framework for dealing with imperfection. We will then show that it is possible to construct a unique scenario for the human, mouse and rat chromosome X that is the closest possible to ultra-perfection.

4.1 Imperfect Sorting

When no ultra-perfect scenario exists, we wish to define a way to score scenarios that are nearly ultra-perfect. There is no easy or straightforward way to do this: the competing parameters include broken common intervals, overlapping inversions, prime intervals and parsimony. In this section, we propose a first measure that is relatively simple to define, and that allows to compare scenarios that would otherwise be difficult to rank.

Our first simplification is that, when trying to build a scenario for two or more species, each common interval should be sorted independently. Since linear

common intervals are rather easy to sort with an ultra-perfect scenario, we focus on the sorting of individual prime common intervals.

A scenario \mathcal{S} between two or more species can be represented as an unrooted tree whose nodes are the genomes, and whose branches are the rearrangement operations. *Removing* an operation r from a scenario \mathcal{S} is done by cutting the branch labeled by r yielding two subtrees called *subscenarios*. We have:

Definition 4. *The imperfection score of a scenario \mathcal{S} is the minimum number of operations that can be removed from \mathcal{S} such that each of the remaining subscenarios is ultra-perfect.*

Our goal is to find, among all possible scenarios with minimum imperfection score, one that is of minimum length.

It turns out that the data on human, mouse, and rat chromosomes X is a very interesting instance of this problem: there are prime common intervals in both the Human-Mouse and the Human-Rat strong interval trees. The prime common interval in the Human-Mouse comparison is maximal, but there is no ultra-perfect scenario since the induced permutation [2], $(\circ -4 \ 6 \ 1 -3 -5 \ 2 \circ)$, does not have a commuting scenario, even with circularization.

The following permutations, obtained from the blocks of [5], model the homologous blocks of the human, mouse, and rat chromosomes X :

$$\begin{aligned} H &= (\circ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \circ) \\ M &= (\circ -6 \ -5 \ 4 \ 13 \ 14 -15 \ 16 \ 1 -3 \ 9 -10 \ 11 \ 12 -7 \ 8 -2 \circ) \\ R &= (\circ -13 \ -4 \ 5 -6 -12 -8 -7 \ 2 \ 1 -3 \ 9 \ 10 \ 11 \ 14 -15 \ 16 \circ) \end{aligned}$$

We first apply single block inversions (they have no impact on the perfection of a scenario) that create an adjacency in one genome that already exists in the other two. There are 5 of them in the chromosome data: {4} and {15} applied to the human chromosome, {7} and {10} applied to the mouse chromosome, and {6} applied to the rat chromosome. The resulting chromosomes are the following, renamed with a subscript that indicates how far the new chromosome is from the original.

$$\begin{aligned} H_{+2} &= (\circ 1 \ 2 \ 3 -4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 -15 \ 16 \circ) \\ M_{+2} &= (\circ -6 \ -5 \ 4 \ 13 \ 14 -15 \ 16 \ 1 -3 \ 9 \ 10 \ 11 \ 12 \ 7 \ 8 -2 \circ) \\ R_{+1} &= (\circ -13 \ -4 \ 5 \ 6 -12 -8 -7 \ 2 \ 1 -3 \ 9 \ 10 \ 11 \ 14 -15 \ 16 \circ) \end{aligned}$$

Next, there are three adjacencies that are shared by the rodents, but are not in the human lineage, and that require inversions longer than single blocks: $(1 -3)$, $(4 \ 13)$ and $(-3 \ 9)$. The corresponding inversions associated to the sets, $\{2, 3\}$, $\{4, 5, 6, 7, 8, 9, 10, 11, 12\}$ and $\{2, 9, 10, 11, 12\}$ can be applied to the H_{+2} genome to yield $H_{+5} = (\circ 1 -3 \ 9 \ 10 \ 11 \ 12 \ 2 -8 -7 -6 -5 \ 4 \ 13 \ 14 -15 \ 16 \circ)$.

In the next section, we will construct an ultra-perfect scenario for the three permutations H_{+5} , M_{+2} and R_{+1} . Since the first two inversions applied to the H_{+2} chromosome are commuting, the imperfection score of the global scenario will be 1, obtained by removing inversion $\{2, 9, 10, 11, 12\}$, which is the best that

can be achieved. The scenario between H_{+2} and H_{+5} is also parsimonious, since it constructs 3 adjacencies present in both the mouse and rat genome, implying that any alternate solution should have the same length. Up to commutation of the two initial inversions, it is easy to show that this is the only solution constructing the 3 adjacencies.

4.2 The Ultra-Perfect Median of Three Genomes

After applying the inversions of the preceding section, the three genomes are the following, where adjacencies common to all three chromosomes are indicated by dots:

$$H_{+5} = (\circ 1 \cdot -3 \cdot 9 \cdot 10 \cdot 11 \ 12 \ 2 \ -8 \cdot -7 \ -6 \cdot -5 \cdot 4 \cdot 13 \ 14 \cdot -15 \cdot 16 \circ)$$

$$M_{+2} = (\circ -6 \cdot -5 \cdot 4 \cdot 13 \ 14 \cdot -15 \cdot 16 \ 1 \cdot -3 \cdot 9 \cdot 10 \cdot 11 \ 12 \ 7 \cdot 8 \ -2 \circ)$$

$$R_{+1} = (\circ -13 \cdot -4 \cdot 5 \cdot 6 \ -12 \ -8 \cdot -7 \ 2 \ 1 \cdot -3 \cdot 9 \cdot 10 \cdot 11 \ 14 \cdot -15 \cdot 16 \circ)$$

It is convenient, at this point, to relabel the blocks so that the remaining differences are more apparent. There are six blocks that we label with respect to the H_{+5} genome order:

$$\overbrace{1 \cdot -3 \cdot 9 \cdot 10 \cdot 11}^1 \overbrace{12}^2 \overbrace{2}^3 \overbrace{-8 \cdot -7}^4 \overbrace{-6 \cdot -5 \cdot 4 \cdot 13}^5 \overbrace{14 \cdot -15 \cdot 16}^6$$

This yields the new representation:

$$H_{+5} = (\circ \mathbf{1 \ 2 \ 3 \ 4 \ 5 \ 6} \circ)$$

$$M_{+2} = (\circ \mathbf{5 \ 6 \ 1 \ 2 \ -4 \ -3} \circ)$$

$$R_{+1} = (\circ \mathbf{-5 \ -2 \ 4 \ 3 \ 1 \ 6} \circ)$$

Given three genomes, deciding if an ultra-perfect scenario connecting them exists begins with a simple check. Indeed, the median of three genomes belongs to all implied pairwise scenarios, thus must share all common intervals of all pairs of genomes. Formally we have:

Proposition 3. *The median M of an ultra-perfect scenario linking three permutations A , B and C contains all common intervals of A and B , of A and C , and of B and C .*

In order to apply Proposition 3 to the mammal chromosomes, we first compute their common intervals:

$$H_{+5} \text{ and } M_{+2}: \{1, 2\}, \{3, 4\}, \{5, 6\}, \{2, 3, 4\}, \{1, 2, 3, 4\}$$

$$H_{+5} \text{ and } R_{+1}: \{3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}, \{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}$$

$$M_{+2} \text{ and } R_{+1}: \{1, 6\}, \{2, 4\}, \{3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}, \{1, 2, 3, 4, 6\}$$

The – unique – permutation that contains all these intervals is a circular chromosome! Its block order, $(5 \ 6 \ 1 \ 2 \ -4 \ -3)$, is the circularization of the mouse genome. It is then a simple exercise to transform the median into each genome. The whole scenario has 7 inversions, 5 of them inverting single blocks; the remaining 2 inversions are $\{3, 4\}$ towards the human chromosome, and $\{2, 3, 4\}$ towards the rat chromosome, see Figure 2.

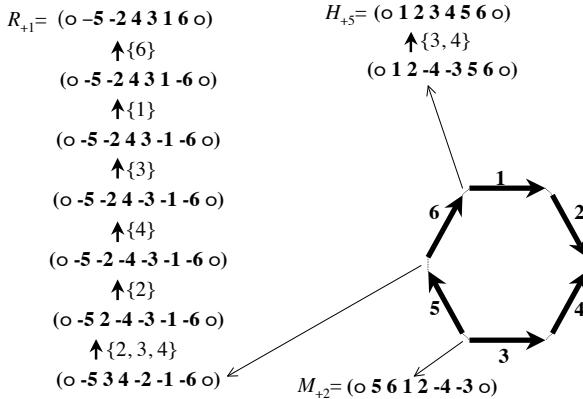


Fig. 2. An ultra-perfect scenario between chromosomes H_{+5} , M_{+2} and R_{+1} . The location where the circular median is cut is shown by thin arrows. The set of inverted blocks is shown between each pair of permutations.

5 Discussion and Conclusions

The search for an ultra-perfect scenario for the human, mouse and rat chromosome X lead to a surprising circular median, deduced by combinatorial techniques. We are certainly not inferring that actual species had circular chromosomes X. The fact that the number of blocks is quite small, $n = 6$, might be the simplest explanation: more than half of the random trios of permutations on six elements have a circular median. However, the remarkable preservation of the circular order of blocks between the human and mouse chromosome X asks for a more satisfying answer. Are there some biological mechanisms that would allow rearrangement operations that preserve a circular order? Among the well known combinatorial operations with this property are the *shift* operation, or a double centromeric inversion.

On the algorithmic side, the circular common interval tree allows us to easily find the set of inversions of an ultra-perfect scenario between two genomes; ultra-perfect DCJ scenarios are essentially ultra-perfect inversion scenarios on a circular version of the genomes. In the case of multiple genomes, the existence and computation of an ultra-perfect scenario should be easy to characterize using the sets of inversions corresponding to pairwise genomes. In the case of nearly ultra-perfect scenario, methods still have to be developed but most efforts will likely lead to hardness results. However, interesting instances of the problem, such as rearrangements between human and rodents, are still quite manageable by manual techniques, and should get easier with the sequencing of additional rodent genomes. It is also relatively easy to score scenarios: the scenario proposed by GRIMM [13] for the human, mouse and rat chromosome X has an imperfection score of 2.

Finally we argue that perfection without ultra-perfection remains, at best, a mathematical exercise. On the other hand, reality is neither perfect, nor ultra-perfect. In this paper we explored some definitions of near ultra-perfection, fully aware that much remains to be done.

References

1. Bérard, S., Bergeron, A., Chauve, C.: Conservation of combinatorial structures in evolution scenarios. In: Lagergren, J. (ed.) RECOMB-WS 2004. LNCS (LNBI), vol. 3388, pp. 1–14. Springer, Heidelberg (2005)
2. Bérard, S., Bergeron, A., Chauve, C., Paul, C.: Perfect sorting by reversals is not always difficult. *IEEE/ACM Trans. Comput. Biology Bioinform.* 4(1), 4–16 (2007)
3. Bérard, S., Chateau, A., Chauve, C., Paul, C., Tannier, E.: Perfect DCJ rearrangement. In: Nelson, C.E., Vialette, S. (eds.) RECOMB-CG 2008. LNCS (LNBI), vol. 5267, pp. 158–169. Springer, Heidelberg (2008)
4. Bergeron, A., Chauve, C., de Montgolfier, F., Raffinot, M.: Computing Common Intervals of k Permutations, with Applications to Modular Decomposition of Graphs. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 779–790. Springer, Heidelberg (2005)
5. Bourque, G., Pevzner, P.A., Tesler, G.: Reconstructing the genomic architecture of ancestral mammals: Lessons from human, mouse, and rat genomes. *Genome Research* 14(4), 507–516 (2004)
6. Braga, M.D., Gautier, C., Sagot, M.-F.: An asymmetric approach to preserve common intervals while sorting by reversals. *Algorithms for Molecular Biology* 4(16) (2009)
7. Figeac, M., Varré, J.-S.: Sorting by reversals with common intervals. In: Jonassen, I., Kim, J. (eds.) WABI 2004. LNCS (LNBI), vol. 3240, pp. 26–37. Springer, Heidelberg (2004)
8. Heber, S., Stoye, J.: Finding all common intervals of k permutations. In: Amir, A., Landau, G.M. (eds.) CPM 2001. LNCS, vol. 2089, pp. 207–218. Springer, Heidelberg (2001)
9. Hsu, W.-L.: PC-trees vs. PQ-trees. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, pp. 207–217. Springer, Heidelberg (2001)
10. Hsu, W.-L., McConnell, R.M.: PC trees and circular-ones arrangements. *Theor. Comput. Sci.* 296(1), 99–116 (2003)
11. Landau, G.M., Parida, L., Weimann, O.: Using PQ trees for comparative genomics. In: Apostolico, A., Crochemore, M., Park, K. (eds.) CPM 2005. LNCS, vol. 3537, pp. 128–143. Springer, Heidelberg (2005)
12. Sagot, M.-F., Tannier, E.: Perfect sorting by reversals. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 42–51. Springer, Heidelberg (2005)
13. Tesler, G.: GRIMM: genome rearrangements web server. *Bioinformatics* 18(3), 492–493 (2002)