On the Approximability of Comparing Genomes with Duplicates

Sébastien Angibaud, Guillaume Fertin, Irena Rusu, Annelyse Thévenin, Stéphane Vialette

On the Approximability of Comparing Genomes with Duplicates

Sébastien Angibaud[1], Guillaume Fertin[1], Irena Rusu[1], Annelyse Thévenin[2], and Stéphane Vialette[3]

[1] Laboratoire d'Informatique de Nantes-Atlantique (LINA), UMR CNRS 6241, Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3, France
{sebastien.angibaud,guillaume.fertin,irena.rusu}@univ-nantes.fr
[2]Laboratoire de Recherche en Informatique (LRI), UMR CNRS 8623, Université Paris-Sud, 91405 Orsay, France
thevenin@lri.fr
[3] IGM-LabInfo, UMR CNRS 8049, Université Paris-Est, 5 Bd Descartes 77454 Marne-la-Vallée, France
vialette@univ-mlv.fr

Abstract. A central problem in comparative genomics consists in computing a (dis-)similarity measure between two genomes, e.g. in order to construct a phylogenetic tree. A large number of such measures has been proposed in the recent past: *number of reversals*, *number of breakpoints*, *number of common* or *conserved intervals* etc. In their initial definitions, all these measures suppose that genomes contain no duplicates. However, we now know that genes can be duplicated within the same genome. One possible approach to overcome this difficulty is to establish a one-to-one correspondence (i.e. a matching) between genes of both genomes, where the correspondence is chosen in order to optimize the studied measure. Then, after a gene relabeling according to this matching and a deletion of the unmatched signed genes, two genomes without duplicates are obtained and the measure can be computed.
In this paper, we are interested in three measures (*number of breakpoints*, *number of common intervals* and *number of conserved intervals*) and three models of matching (*exemplar*, *intermediate* and *maximum matching* models). We prove that, for each model and each measure $\mathcal{M}$, computing a matching between two genomes that optimizes $\mathcal{M}$ is **APX**–hard. We show that this result remains true even for two genomes G_1 and G_2 such that G_1 contains no duplicates and no gene of G_2 appears more than twice. Therefore, our results extend those of [7, 10, 13]. Besides, in order to evaluate the possible existence of approximation algorithms concerning the number of breakpoints, we also study the complexity of the following decision problem: is there an exemplarization (resp. an intermediate matching, a maximum matching) that induces no breakpoint ? In particular, we extend a result of [13] by proving the problem to be **NP**–complete in the exemplar model for a new class of instances, we note that the problems are equivalent in the intermediate and the exemplar models and we show that the problem is in **P** in the maximum matching model. Finally, we focus on a fourth measure, closely related to the number of breakpoints: the *number of adjacencies*, for which we give several constant ratio approximation algorithms in the maximum matching model, in the case where genomes contain the same number of duplications of each gene.

Keywords: genome rearrangements, **APX**–hardness, duplicate genes, breakpoints, adjacencies, common intervals, conserved intervals, approximation algorithms.

1 Introduction and Preliminaries

In comparative genomics, computing a measure of (dis-)similarity between two genomes is a central problem: such a measure can be used, for instance, to construct phylogenetic trees. The measures defined so far essentially fall into two categories: the first one consists in counting the minimum number of operations needed to transform a genome into another (e.g. the *edit distance* [21] or the *number of reversals* [4]). The second one contains (dis-)similarity measures based on the genome structure, such as the *number of breakpoints* [7], the *conserved intervals distance* [6], the *number of common intervals* [10], *SAD* (Sum Adjacency Disruption) and *MAD* (Maximum Adjacency Disruption) [24] etc.

When genomes contain no duplicates, most measures can be computed in polynomial time. However, assuming that genomes contain no duplicates is too limited. Indeed, it has been recently shown that a great number of duplicates exists in some genomes. For example, in [20], authors estimate that 15% of genes are duplicated in the human genome. A possible approach to overcome this difficulty is to specify a one-to-one correspondence (i.e. a *matching*) between genes of two genomes and to remove the unmatched genes, thus obtaining two genomes with identical gene content and no duplicates. Usually, the above mentioned matching is chosen in order to optimize the studied measure, following the parsimony principle. Three models achieving this correspondence have been proposed : the *exemplar* model [23], the *intermediate* model [3] and the *maximum matching* model [25]. Before defining precisely the measures and models studied in this paper, we need to introduce some notation.

Notation used in the paper. Let $\mathcal{F}$ be a set of *genes*, where each gene is represented by an unsigned integer. A genome G is a string composed of signed integers (called *signed genes*), where the sign indicates the orientation of the gene inside the genome. For any genome G, we denote by $\mathcal{F}_G$ the set of unsigned integers (genes) that are present in G. For any signed gene g, let $-g$ be the signed gene having the opposite sign and let $|g| \in \mathcal{F}_G$ be the corresponding (unsigned) gene.

Given a genome G without duplicates (i.e. without signed genes with the same absolute value) and two signed genes a, b such that a is located before b, let $G[a, b]$ be the set $S \subseteq \mathcal{F}_G$ of genes located between a and b in G, $|a|$ and $|b|$ included. We also denote by $[a, b]_G$ the substring of G starting at a and finishing at b in G.

Denote by n_G the size of genome G, that is the number of signed genes it contains. Let $G[p]$, $1 \leq p \leq n_G$, be the signed gene that occurs at position p on genome G. Let $N_G[p]$, $1 \leq p \leq n_G$, be the number of occurrences of $|G[p]|$ in the first $(p-1)$ positions of G and, given a gene g, let $\text{occ}(g, G)$ be the number of occurrences of g in G. Let $\text{occ}(G) = \max\{\text{occ}(g, G) | g \in \mathcal{F}_G\}$. A pair of genomes ($G_1$,$G_2$) is said to be *of type* (x, y) if $\text{occ}(G_1) = x$ and $\text{occ}(G_2) = y$. A pair of genomes (G_1,G_2) is said to be *balanced* if, for each gene $g \in \mathcal{F}_{G_1} \cup \mathcal{F}_{G_2}$, we have $\text{occ}(g, G_1) = \text{occ}(g, G_2)$ (otherwise, (G_1,G_2) will be said to be *unbalanced*). Note that a pair (G_1, G_2) of type (x, x) is not necessary balanced.

We define a *duo* in a genome G as a pair of successive signed genes. Given a duo $d_i = (G[i], G[i+1])$ in a genome G, we note by $-d_i$ the duo equal to $(-G[i+1], -G[i])$. Let (d_1, d_2) be a pair of duos ; (d_1, d_2) is called a *duo match* if d_1 is a duo of G_1, d_2 is a duo of G_2, and if either $d_1 = d_2$ or $d_1 = -d_2$.

For example, consider the genome $G_1 = +1\ +2\ +3\ +4\ +5\ -1\ -2\ +6\ -2$. Then, $\mathcal{F}_G = \{1, 2, 3, 4, 5, 6\}$, $n_{G_1} = 9$, $\text{occ}(1, G_1) = 2$, $\text{occ}(G_1) = 3$, $G_1[7] = -2$, $-G_1[7] = +2$, $|G_1[7]| = 2$ and $N_{G_1}[7] = 1$. Let G_2 be the genome $G_2 = +2\ -1\ +6\ +3\ -5\ -4\ +2\ -1\ -2$. Then the pair ($G_1$, G_2) is balanced and is of type $(3, 3)$. Let $d_1 = (G_1[4], G_1[5])$ be the duo $(+4, +5)$ and d_2 be the duo $(G_2[5], G_2[6])$. The pair (d_1, d_2) is a duo match. Now, consider the genome $G_3 = +3-2+6+4-1+5$ without duplicates. We have $G_3[+6, -1] = \{1, 4, 6\}$ and $[+6, -1]_{G_3} = (+6, +4, -1)$.

Breakpoints, adjacencies, common and conserved intervals. Let us now define the four measures we will study in this paper. Let G_1 and G_2 be two genomes without duplicates and with the same gene content, that is $\mathcal{F}_{G_1} = \mathcal{F}_{G_2}$.

Breakpoint and Adjacency. Let (a, b) be a duo in G_1. We say that the duo (a, b) induces a *breakpoint* of (G_1, G_2) if neither (a, b) nor $(-b, -a)$ is a duo in G_2. Otherwise, we say that (a, b) induces an *adjacency* of (G_1, G_2). For example, when $G_1 = +1 + 2 + 3 + 4 + 5$ and $G_2 = +5 -$

$4-3+2+1$, the duo $(2,3)$ in G_1 induces a breakpoint of (G_1, G_2) while $(3,4)$ in G_1 induces an adjacency of (G_1, G_2). We note by $B(G_1, G_2)$ (resp. $A(G_1, G_2)$) the number of breakpoints (resp. the number of adjacencies) that exist between G_1 and G_2.

Common interval. A *common interval* of (G_1, G_2) is a substring of G_1 such that G_2 contains a permutation of this substring (not taking signs into account). For example, consider $G_1 = +1+2+3+4+5$ and $G_2 = +2-4+3+5+1$. The substring $[+3,+5]_{G_1}$ is a common interval of (G_1, G_2).

Conserved interval. Consider two signed genes a and b of G_1 such that a precedes b, where the precedence relation is extensive in the sense that, possibly, $a = b$. The substring $[a,b]_{G_1}$ is a *conserved interval* of (G_1, G_2) if either (i) a precedes b in the string G_2 and $G_2[a,b] = G_1[a,b]$, or (ii) $-b$ precedes $-a$ and $G_2[-b,-a] = G_1[a,b]$. For example, if $G_1 = +1+2+3+4+5$ and $G_2 = -5-4+3-2+1$, the substring $[+2,+5]_{G_1}$ is a conserved interval of (G_1, G_2) (indeed, the signed gene -2 precedes -5 in G_2 and $G_2[-5,-2] = G_1[+2,+5]$). We note that a conserved interval is actually a common interval, but with additional restrictions on its extremities.

Dealing with duplicates in genomes. When genomes contain duplicates, we cannot directly compute the measures defined in the previous paragraph. A solution consists in finding a one-to-one correspondence (i.e. a matching) between duplicated genes of G_1 and G_2 ; we then use this correspondence to rename genes of G_1 and G_2, and we delete the unmatched signed genes in order to obtain two genomes G_1' and G_2' such that G_2' is a *permutation* of G_1' ; thus, the measure computation becomes possible. In this paper, we will focus on three models of matching : the *exemplar*, *intermediate* and *maximum matching* models.

- The *exemplar model* [23]: for each gene g, we keep in the matching $\mathcal{M}$ only one occurrence of g in G_1 and in G_2, and we remove all the other occurrences. Hence, we obtain two genomes G_1^E and G_2^E without duplicates. The triplet $(G_1^E, G_2^E, \mathcal{M})$ is called an *exemplarization* of (G_1, G_2). Note that in this model, $\mathcal{M}$ can be inferred from the exemplarized genomes G_1^E and G_2^E. Thus, in the rest of the paper, any exemplarization $(G_1^E, G_2^E, \mathcal{M})$ of (G_1, G_2) will be only described by the pair (G_1^E, G_2^E).
- The *intermediate model* [3]: in this model, for each gene g, we keep in the matching $\mathcal{M}$ an arbitrary number k_g, $1 \leq k_g \leq min(occ(g, G_1), occ(g, G_2))$, in order to obtain genomes G_1^I and G_2^I. We call the triplet $(G_1^I, G_2^I, \mathcal{M})$ an *intermediate matching* of (G_1, G_2).
- The *maximum matching model* [25]: in this case, we keep in the matching $\mathcal{M}$ the maximum number of signed genes in both genomes. More precisely, we look for a one-to-one correspondence between signed genes of G_1 and G_2 that matches, for each gene g, exactly $min(\mathrm{occ}(g, G_1), \mathrm{occ}(g, G_2))$ occurrences. After this operation, we delete each unmatched signed gene. The triplet $(G_1^M, G_2^M, \mathcal{M})$ obtained by this operation is called a *maximum matching* of (G_1, G_2).

Problems studied in this paper. Consider two genomes G_1 and G_2 with duplicates. Let ECOMI (resp. ICOMI, MCOMI) be the problem which consists in finding an exemplarization (resp. intermediate matching, maximum matching) $(G_1', G_2', \mathcal{M})$ of (G_1, G_2) such that the number of common intervals of (G_1', G_2') is maximized. Moreover, let ECONSI (resp. ICONSI, MCONSI) be the problem which consists in finding an exemplarization (resp. intermediate matching, maximum matching) $(G_1', G_2', \mathcal{M})$ of (G_1, G_2) such that the number of conserved intervals of $(G_1', G_2', \mathcal{M})$ is maximized. In Section 2, we prove the **APX**–hardness [22] of ECOMI and ECONSI, even for genomes G_1 and G_2 such that $\mathrm{occ}(G_1) = 1$ and $\mathrm{occ}(G_2) = 2$. These results induce the **APX**–hardness under the other models (i.e., ICOMI, MCOMI, ICONSI and MCONSI are **APX**–hard). These results extend in particular those of [7, 10].

Let EBD (resp. IBD, MBD) be the problem which consists in finding an exemplarization (resp. intermediate matching, maximum matching) $(G'_1, G'_2, \mathcal{M})$ of (G_1, G_2) that minimizes the number of breakpoints between G'_1 and G'_2. In Section 3, we prove the **APX**–hardness of EBD, even for genomes G_1 and G_2 such that $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$. This result implies that IBD and MBD are also **APX**–hard, and extends those of [13].

Let ZEBD (resp. ZIBD, ZMBD) be the problem which consists in determining, for two genomes G_1 and G_2, whether there exists an exemplarization (resp. intermediate matching, maximum matching) which induces *zero breakpoint*. In section 4, we study the complexity of ZEBD, ZMBD and ZIBD: in particular, we extend a result of [13] by proving ZEBD to be **NP**–complete for a new class of instances. We also note that the problems ZEBD and ZIBD are equivalent, and we show that ZMBD is in **P**.

Finally, in Section 5, we focus on a fourth measure, closely related to the number of breakpoints: the *number of adjacencies*, for which we give several constant ratio approximation algorithms in the maximum matching model, in the case where genomes are balanced.

2 EComI and EConsI are APX–hard

Consider two genomes G_1 and G_2 with duplicates, and let ECOMI (resp. ICOMI, MCOMI) be the problem which consists in finding an exemplarization (resp. intermediate matching, maximum matching) $(G'_1, G'_2, \mathcal{M})$ of (G_1, G_2) such that the number of common intervals of (G'_1, G'_2) is maximized. Moreover, let ECONSI (resp. ICONSI, MCONSI) be the problem which consists in finding an exemplarization (resp. intermediate matching, maximum matching) $(G'_1, G'_2, \mathcal{M})$ of (G_1, G_2) such that the number of conserved intervals of $(G'_1, G'_2, \mathcal{M})$ is maximized.

ECOMI and MCOMI have been proved to be **NP**–complete even if $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$ in [10]. Besides, in [6], Blin and Rizzi have studied the problem of computing a distance built on the number of conserved intervals. This distance differs from the number of conserved intervals we study in this paper, mainly in the sense that (i) it can be applied to two *sets* of genomes (as opposed to two genomes in our case), and (ii) the distance between two identical genomes of length n is equal to 0 (as opposed to $\frac{n(n+1)}{2}$ in our case). Blin and Rizzi [6] proved that finding the minimum distance is **NP**–complete, under both the exemplar and maximum matching models. A closer analysis of their proof shows that it can be easily adapted to prove that ECONSI and MCONSI are **NP**–complete, even in the case $\text{occ}(G_1) = 1$.

We can conclude from the above results that ICOMI and ICONSI are also **NP**–complete, since when one genome contains no duplicates, *exemplar*, *intermediate* and *maximum matching* models are equivalent.

In this section, we improve the above results by showing that the six problems ECOMI, ICOMI, MCOMI, ECONSI, ICONSI and MCONSI are **APX**–hard, even when genomes G_1 and G_2 are such that $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$. The main result is Theorem 1, which will be completed by Corollary 1 at the end of the section.

Theorem 1. ECOMI *and* ECONSI *are* **APX***–hard even when genomes* G_1 *and* G_2 *are such that* $\text{occ}(G_1) = 1$ *and* $\text{occ}(G_2) = 2$.

We prove Theorem 1 by using an *L-reduction* [22] from the MIN-VERTEX-COVER problem on cubic graphs, denoted here by MIN-VERTEX-COVER-3. Let $G = (V, E)$ be a cubic graph, i.e. for all $v \in V, degree(v) = 3$. A set of vertices $V' \subseteq V$ is called a *vertex cover* of G if for each edge $e \in E$,

there exists a vertex $v \in V'$ such that e is incident to v. The problem MIN-VERTEX-COVER-3 is defined as follows:

Problem: MIN-VERTEX-COVER-3
Input: A cubic graph $G = (V, E)$.
Solution: A vertex cover V' of G.
Measure: The cardinality of V'.

MIN-VERTEX-COVER-3 was proved to be **APX**–complete in [1].

2.1 Reduction

Let $G = (V, E)$ be an instance of MIN-VERTEX-COVER-3, where G is a cubic graph with $V = \{v_1 \dots v_n\}$ and $E = \{e_1 \dots e_m\}$. Consider the transformation R which associates to the graph G two genomes G_1 and G_2 in the following way, where each gene has a positive sign.

$$G_1 = b_1\ b_2 \dots b_m\ x\ a_1\ C_1\ f_1\ a_2\ C_2\ f_2 \dots a_n\ C_n\ f_n\ y\ b_{m+n}, b_{m+n-1} \dots b_{m+1} \quad (1)$$

$$G_2 = y\ a_1\ D_1\ f_1\ b_{m+1}\ a_2\ D_2\ f_2\ b_{m+2} \dots b_{m+n-1}\ a_n\ D_n\ f_n\ b_{m+n}\ x \quad (2)$$

with :

- for each i, $1 \leq i \leq n, a_i = 6i - 5$, $f_i = 6i$
- for each i, $1 \leq i \leq n, C_i = (a_i + 1), (a_i + 2), (a_i + 3), (a_i + 4)$
- for each i, $1 \leq i \leq n + m, b_i = 6n + i$
- $x = 7n + m + 1$ and $y = 7n + m + 2$
- for each i, $1 \leq i \leq n, D_i = (a_i + 3), (b_{j_i}), (a_i + 1), (b_{k_i}), (a_i + 4), (b_{l_i}), (a_i + 2)$ where e_{j_i}, e_{k_i} and e_{l_i} are the edges which are incident to v_i in G, with $j_i < k_i < l_i$.

In the following, genes b_i, $1 \leq i \leq m$, are called *markers*. There is no duplicated gene in G_1 and the markers are the only duplicated genes in G_2 ; these genes occur twice in G_2. Hence, we have $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$.

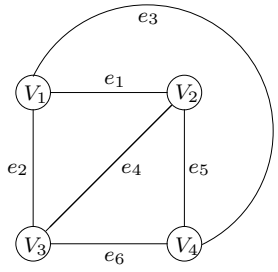

Fig. 1. The cubic graph G.

To illustrate the reduction, consider the cubic graph G of Figure 1. From G, we construct the following genomes G_1 and G_2:

$$\underbrace{25\ 26\ 27\ 28\ 29\ 30\ 35}_{b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ x}\ 1\ \underbrace{2\ 3\ 4\ 5}_{C_1}\ 6\ 7\ \underbrace{8\ 9\ 10\ 11}_{C_2}\ 12\ 13\ \underbrace{14\ 15\ 16\ 17}_{C_3}\ 18\ 19\ \underbrace{20\ 21\ 22\ 23}_{C_4}\ 24\ \underbrace{36}_{y}\ \underbrace{34}_{b_{10}}\ \underbrace{33}_{b_9}\ \underbrace{32}_{b_8}\ \underbrace{31}_{b_7}$$

$$\underbrace{36}_{y}\ 1\ \underbrace{4\ 25\ 2\ 26\ 5\ 27\ 3}_{D_1}\ 6\ \underbrace{31}_{b_7}\ 7\ \underbrace{10\ 25\ 8\ 28\ 11\ 29\ 9}_{D_2}\ 12\ \underbrace{32}_{b_8}\ 13\ \underbrace{16\ 26\ 14\ 28\ 17\ 30\ 15}_{D_3}\ 18\ \underbrace{33}_{b_9}\ 19\ \underbrace{22\ 27\ 20\ 29\ 23\ 30\ 21}_{D_4}\ 24\ \underbrace{34}_{b_{10}}\ \underbrace{35}_{x}$$

2.2 Preliminary results

In order to prove Theorem 1, we first give four intermediate lemmas. In the following, a common interval for the ECOMI problem or a conserved interval for ECONSI is called a *robust interval*. Besides, a *trivial interval* will denote either an interval of length one (i.e. a singleton), or the whole genome.

Lemma 1. *For any exemplarization (G_1, G_2^E) of (G_1, G_2), the non trivial robust intervals of (G_1, G_2^E) are necessarily contained in some sequence $a_i C_i f_i$ of G_1 $(1 \leq i \leq n)$.*

Proof. We start by proving the lemma for common intervals, and we will then extend it to conserved intervals. First, we prove that, for any exemplarization (G_1, G_2^E) of (G_1, G_2), each common interval I such that $|I| \geq 2$ contains either both of x, y or none of them. This further implies that I covers the whole genome. Suppose there exists a common interval I_x (recall that by definition I_x is on G_1) such that $|I_x| \geq 2$ and I_x contains x. Let PI_x be the permutation of I_x in G_2^E. The interval I_x must contain either b_m or a_1. Let us detail each of the two cases:

(a) If I_x contains b_m, then PI_x contains b_m too. Notice that there is some i, $1 \leq i \leq n$, such that b_m belongs to D_i in G_2^E. Then PI_x contains all genes between D_i and x in G_2^E. Thus PI_x contains b_{m+n}. Consequently, I_x contains b_{m+n} and it also contains y.
(b) If I_x contains a_1, then PI_x contains a_1 too. Then PI_x contains all genes between a_1 and x. Thus PI_x contains b_{m+n}. Hence, I_x contains b_{m+n} and then it also contains y.

Now, suppose that I_y is a common interval such that $|I_y| \geq 2$ and I_y contains y. Let PI_y be the permutation of I_y on G_2^E. The interval I_y must contain either b_{m+n} or f_n. Let us detail each of the two cases:

(a) If I_y contains b_{m+n}, then PI_y contains b_{m+n} too. Thus PI_y contains all genes between b_{m+n} and y. Hence PI_y contains all the sequences D_i, $1 \leq i \leq n$. In particular, PI_y contains all the markers and consequently I_y must contain x.
(b) If I_y contains f_n, then PI_y contains f_n too. Then PI_y contains all genes between f_n and y. In particular, PI_y contains b_{m+n-1} and then I_y contains b_{m+n-1} too. Hence, I_y also contains b_{m+n}, similarly to the previous case. Thus I_y contains x.

We conclude that each non singleton common interval containing either x or y necessarily contains both x and y. Therefore, and by construction of G_2, there is only one such interval, that is G_1 itself. Hence, any non trivial common interval is necessarily, in G_1, either strictly on the left of x, or between x and y, or strictly on the right of y. Let us analyze these different cases:

- Let I be a non trivial common interval situated strictly on the left of x in G_1. Thus I is a sequence of at least two consecutive markers. Since in any exemplarization (G_1, G_2^E) of (G_1, G_2), every marker in G_2^E has neighboring genes which are not markers, this contradicts the fact that I is a common interval.
- Let I be a non trivial common interval situated strictly on the right of y in G_1. Then I is a substring of $b_{m+n}, \ldots, b_{m+1}$ containing at least two genes. In any exemplarization (G_1, G_2^E) of (G_1, G_2), for each pair (b_{m+i}, b_{m+i+1}) of G_2^E, with $1 \leq i < n$, we have $a_{i+1} \in G_2^E[b_{m+i}, b_{m+i+1}]$. This contradicts the fact that I is strictly on the right of y in G_1.

– Let I be a non trivial common interval lying between x and y in G_1. For any exemplarization (G_1, G_2^E) of (G_1, G_2), a common interval cannot contain, in G_1, both f_i and a_{i+1} for some i, $1 \leq i \leq n-1$ (since b_{m+i} is situated between f_i and a_{i+1} in G_2^E and on the right of x in G_1). Hence, a non trivial common interval of (G_1, G_2^E) is included in some sequence $a_i C_i f_i$ in G_1, $1 \leq i \leq n$.

This proves the lemma for common intervals. By definition, any conserved interval is necessarily a common interval. So, a non trivial conserved interval of (G_1, G_2^E) is included in some sequence $a_i C_i f_i$ in G_1, $1 \leq i \leq n$. The lemma is proved. □

Lemma 2. *Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and $i \in [1 \dots n]$. Let Δ_i be a substring of $[a_i + 3, a_i + 2]_{G_2^E}$ that does not contain any marker. If $|\Delta_i| \in \{2, 3\}$, then there is no robust interval I of (G_1, G_2^E) such that Δ_i is a permutation of I.*

Proof. First, we prove that there is no permutation I of Δ_i such that I is a common interval of (G_1, G_2^E). Next, we show that there is no permutation I of Δ_i such that I is a conserved interval. By Lemma 1, we know that a non trivial common interval of (G_1, G_2^E) is a substring of some sequence $a_i C_i f_i$, $1 \leq i \leq n$. This substring contains only consecutive integers. Therefore, if there exists a permutation I of Δ_i such that I is a common interval of (G_1, G_2^E), then Δ_i must be a permutation of consecutive integers. If $|\Delta_i| = 2$, we have $\Delta_i = (p, q)$ where p and q are not consecutive integers and if $|\Delta_i| = 3$, then we have $\Delta_i = (a_i + 3, a_i + 1, a_i + 4)$ or $\Delta_i = (a_i + 1, a_i + 4, a_i + 2)$. In these three cases, Δ_i is not a permutation of consecutive integers. Hence, there is no permutation I of Δ_i such that I is a common interval of (G_1, G_2^E). Moreover, any conserved interval is also a common interval. Thus, there is no permutation I of Δ_i such that I is a conserved interval of (G_1, G_2^E). □

For more clarity, let us now introduce some notation. Given a graph $G = (V, E)$, let $VC = \{v_{i_1}, v_{i_2} \dots v_{i_k}\}$ be a vertex cover of G. Let $R(G) = (G_1, G_2)$ be the pair of genomes defined by the construction described in (1) and (2). Now, let F be the function which associates to VC, G_1 and G_2 an exemplarization $F(VC)$ of (G_1, G_2) as follows. In G_2, all the markers are removed from the sequences D_i for all $i \neq i_1, i_2 \dots i_k$. Next, for each marker which is still present twice, one of its occurrences is arbitrarily removed. Since in G_2 only markers are duplicated, we conclude that $F(VC)$ is an exemplarization of (G_1, G_2).

Given a cubic graph G and genomes G_1 and G_2 obtained by the transformation $R(G)$, let us define the function S which associates to an exemplarization (G_1, G_2^E) of (G_1, G_2) the vertex cover VC of G defined as follows: $VC = \{v_i | 1 \leq i \leq n \wedge \exists j \in \{1 \dots m\}, b_j \in G_2^E[a_i, f_i]\}$. In other words, we keep in VC the vertices v_i of G for which there exists some gene b_j such that b_j is in $G_2^E[a_i, f_i]$. We now prove that VC is a vertex cover. Consider an edge e_p of G. By construction of G_1 and G_2, there exists some i, $1 \leq i \leq n$, such that gene b_p is located between a_i and f_i in G_2^E. The presence of gene b_p between a_i and f_i implies that vertex v_i belongs to VC. We conclude that each edge is incident to at least one vertex of VC.

Let W be the function defined on $\{\text{ECONSI}, \text{ECOMI}\}$ by $W(\text{pb}) = 1$ if $\text{pb} = \text{ECONSI}$ and $W(\text{pb}) = 4$ if $\text{pb} = \text{ECOMI}$. Let $\mathsf{opt}_{pb}(A)$ be the optimum result of an instance A for an optimization problem pb, $\text{pb} \in \{\text{ECOMI}, \text{ECONSI}, \text{MIN-VERTEX-COVER-3}\}$.

We now define the function T whose arguments are a problem $\text{pb} \in \{\text{ECONSI}, \text{ECOMI}\}$ and a cubic graph G. Let $R(G) = (G_1, G_2^E)$ as usual. Then $T(\text{pb}, G)$ is defined as the number of robust trivial intervals of (G_1, G_2^E) with respect to pb. Let n and m be respectively the number of vertices

and the number of edges of G. We have $T(\text{ECONSI}, G) = 7n+m+2$ and $T(\text{ECOMI}, G) = 7n+m+3$. Indeed, for ECOMI, there are $7n+m+2$ singletons and we also need to consider the whole genome.

Lemma 3. *Let $pb \in \{\text{ECOMI}, \text{ECONSI}\}$. Let G be a cubic graph and $R(G) = (G_1, G_2)$. Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and let i, $1 \leq i \leq n$. Then only two cases can occur with respect to D_i.*

1. *Either in G_2^E, all the markers from D_i were removed, and in this case, there are exactly $W(pb)$ non trivial robust intervals involving D_i.*
2. *Or in G_2^E, at least one marker was kept in D_i, and in this case, there is no non trivial robust interval involving D_i.*

Proof. We first prove the lemma for the ECOMI problem and then we extend it to ECONSI. Lemma 1 implies that each non trivial common interval I of (G_1, G_2^E) is contained in some substring of $a_iC_if_i$, $1 \leq i \leq n$. So, the permutation of I on G_2^E is contained in a substring of $a_iD_if_i$, $1 \leq i \leq n$.

Consider i, $1 \leq i \leq n$, and suppose that all the markers from D_i are removed on G_2^E. Thus, $a_iC_if_i$, C_i, a_iC_i and C_if_i are common intervals of (G_1, G_2^E). Let us now show that there is no other non trivial common interval involving D_i. Let Δ_i be a substring of $[a_i+3, a_i+2]_{G_2^E}$ such that $|\Delta_i| \in \{2, 3\}$. By Lemma 2, we know that Δ_i is not a common interval. The remaining intervals are (a_i, a_i+3), (a_i, a_i+3, a_i+1), $(a_i, a_i+3, a_i+1, a_i+4)$, $(a_i+1, a_i+4, a_i+2, f_i)$, (a_i+4, a_i+2, f_i) and (a_i+2, f_i). By construction, none of them can be a common interval, because none of them is a permutation of consecutive integers. Hence, there are only four non trivial common intervals involving D_i in G_2^E. Among these four common intervals, only $a_iC_if_i$ is a conserved interval too. In the end, if all the markers are removed from D_i, there are exactly four non trivial common intervals and one non trivial conserved interval involving D_i. So, given a problem $\text{pb} \in \{\text{ECOMI}, \text{ECONSI}\}$, there are exactly $W(\text{pb})$ non trivial robust intervals involving D_i.

Now, suppose that at least one marker of D_i is kept in G_2^E. Lemma 1 shows that each non trivial common interval I of (G_1, G_2^E) is contained in some substring of $a_iC_if_i$, $1 \leq i \leq n$. Since no marker is present in a sequence $a_iC_if_i$, we deduce that there does not exist any trivial common interval containing a marker. So, a non trivial common interval involving D_i only must contain a substring Δ_i of $[a_i+3, a_i+2]_{G_2^E}$ such that Δ_i contains no marker. Since no marker is an extremity of $[a_i+3, a_i+2]_{G_2^E}$, we have $|\Delta_i| \leq 3$. By Lemma 2, we know that Δ_i is not a common interval. The remaining intervals to be considered are the intervals $a_i\Delta_i$ and Δ_if_i. By construction of $a_iC_if_i$, these intervals are not common intervals (the absence of gene a_i+2 for $a_i\Delta_i$ and of gene a_i+3 for Δ_if_i implies that these intervals are not a permutation of consecutive integers). Hence, these intervals cannot be conserved intervals either. □

Lemma 4. *Let $pb \in \{\text{ECOMI}, \text{ECONSI}\}$. Let $G = (V, E)$ be a cubic graph with $V = \{v_1 \dots v_n\}$ and $E = \{e_1 \dots e_m\}$ and let G_1, G_2 be the two genomes obtained by $R(G)$.*

1. *Let VC be a vertex cover of G and denote $k = |VC|$. Then the exemplarization $F(VC)$ of (G_1, G_2) has at least $N = n\,W(pb) + T(pb, G) - W(pb) \cdot k$ robust intervals.*
2. *Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and let VC' be the vertex cover of G obtained by $S(G_1, G_2^E)$. Then $|VC'| = \frac{W(pb)\cdot n + T(pb,G) - N}{W(pb)}$, where N is the number of robust intervals of (G_1, G_2^E).*

Proof. 1. Let $\text{pb} \in \{\text{ECOMI}, \text{ECONSI}\}$. Let G be a cubic graph and let G_1 and G_2 be the two genomes obtained by $R(G)$. Let VC be a vertex cover of G and denote $k = |VC|$. Let (G_1, G_2^E) be the

exemplarization of (G_1, G_2) obtained by $F(VC)$. By construction, we have at least $(n-k)$ substrings D_i in G_2^E for which all the markers are removed. By Lemma 3, we know that each of these substrings implies the existence of $W(\text{pb})$ non trivial robust intervals. So, we have at least $W(\text{pb})(n-k)$ non trivial robust intervals. Moreover, by definition of $T(\text{pb}, G)$, the number of trivial robust intervals of (G_1, G_2^E) is exactly $T(\text{pb}, G)$. Thus, we have at least $N = W(\text{pb}) \cdot n + T(\text{pb}, G) - W(\text{pb}) \cdot k$ robust intervals of (G_1, G_2^E).

2. Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and let $n-j$ be the number of sequences D_i, $1 \leq i \leq n$, for which all markers have been deleted in G_2^E. Then, by Lemmas 1 and 3, the number of robust intervals of (G_1, G_2^E) is equal to $N = W(\text{pb}) \cdot n + T(\text{pb}, G) - W(\text{pb}) \cdot j$. Let VC' be the vertex cover obtained by $S(G_1, G_2^E)$. Each marker has one occurrence in G_2^E and these occurrences lie in j sequences D_i. So, by definition of S, we conclude that $|VC'| = j = \frac{W(\text{pb}) \cdot n + T(\text{pb}, G) - N}{W(\text{pb})}$. □

2.3 Main result

Let us first define the notion of *L-reduction* [22]: let A and B be two optimization problems and c_A, c_B be respectively their cost functions. An *L-reduction* from problem A to problem B is a pair of polynomial-time computable functions R and S with the following properties:

(a) If x is an instance of A, then $R(x)$ is an instance of B ;
(b) If x is an instance of A and y is a solution of $R(x)$, then $S(y)$ is a solution of A ;
(c) If x is an instance of A and $R(x)$ is its corresponding instance of B, then there is some positive constant α such that $\mathsf{opt}_B(R(x)) \leq \alpha.\mathsf{opt}_A(x)$;
(d) If s is a solution of $R(x)$, then there is some positive constant β such that $|\mathsf{opt}_A(x) - c_A(S(s))| \leq \beta|\mathsf{opt}_B(R(x)) - c_B(s)|$.

We prove Theorem 1 by showing that the pair (R, S) defined previously is an *L-reduction* from Min-Vertex-Cover-3 to EConsI and from Min-Vertex-Cover-3 to EComI. First note that properties (a) and (b) are obviously satisfied by R and S.

Consider $\text{pb} \in \{\text{EComI}, \text{EConsI}\}$. Let $G = (V, E)$ be a cubic graph with n vertices and m edges. We now prove properties (c) and (d). Consider the genomes G_1 and G_2 obtained by $R(G)$. For sake of clarity, we abbreviate here and in the following $\mathsf{opt}_{\text{Min-Vertex-Cover-3}}$ to $\mathsf{opt}_{\text{Min-VC}}$. First, we need to prove that there exists $\alpha \geq 0$ such that $\mathsf{opt}_{\text{pb}}(G_1, G_2) \leq \alpha.\mathsf{opt}_{\text{Min-Vertex-Cover-3}}(G)$.

Since G is cubic, we have the following properties:

$$n \geq 4 \tag{3}$$

$$m = \frac{1}{2}\sum_{i=1}^{n} degree(v_i) = \frac{3n}{2} \tag{4}$$

$$\mathsf{opt}_{\text{Min-VC}}(G) \geq \frac{m}{3} = \frac{n}{2} \tag{5}$$

To explain property (5), remark that, in a cubic graph G with n vertices and m edges, each vertex covers three edges. Thus, a set of k vertices covers at most $3k$ edges. Hence, any vertex cover of G must contain at least $\frac{m}{3}$ vertices.

By Lemma 3, we know that sequences of the form $a_i C_i f_i$, $1 \leq i \leq n$, contain either zero or $W(\text{pb})$ non trivial robust intervals. By Lemma 1, there are no other non trivial robust intervals. So, we have the following inequality:

$$\mathsf{opt}_{\text{pb}}(G_1, G_2) \leq \underbrace{T(\text{pb}, G)}_{\textit{trivial robust intervals}} + W(\text{pb}) \cdot n$$

If pb = ECOMI, we have:

$$\begin{aligned} &\mathsf{opt}_{\text{ECOMI}}(G_1, G_2) \leq 7n + m + 3 + 4n \\ &\mathsf{opt}_{\text{ECOMI}}(G_1, G_2) \leq \frac{27n}{2} \quad \textit{by (3) and (4)} \end{aligned} \tag{6}$$

And if pb = ECONSI, we have :

$$\begin{aligned} &\mathsf{opt}_{\text{ECONSI}}(G_1, G_2) \leq 7n + m + 2 + n \\ &\mathsf{opt}_{\text{ECONSI}}(G_1, G_2) \leq \frac{21n}{2} \quad \textit{by (3) and (4)} \end{aligned} \tag{7}$$

Altogether, by (5), (6) and (7), we prove property (c) with $\alpha = 27$.

Now, let us prove property (d). Let $VC = \{v_{i_1}, v_{i_2} \dots v_{i_P}\}$ be a minimum vertex cover of G. Then $P = \mathsf{opt}_{\text{MIN-VC}}(G)$. Let G_1 and G_2 be the genomes obtained by $R(G)$. Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and let k' be the number of robust intervals of (G_1, G_2^E). Finally, let VC' be the vertex cover of G such that $VC' = S(G_1, G_2^E)$. We need to find a positive constant β such that $|P - |VC'|| \leq \beta|\mathsf{opt}_{\text{pb}}(G_1, G_2) - k'|$.

For pb $\in \{\text{ECOMI}, \text{ECONSI}\}$, let N_{pb} be the number of robust intervals between the two genomes obtained by $F(VC)$. By the first property of Lemma 4, we have

$$\mathsf{opt}_{\text{pb}}(G_1, G_2) \geq N_{\text{pb}} \geq W(\text{pb}) \cdot n + T(\text{pb}, G) - W(\text{pb}) \cdot P$$

So, it is sufficient to prove that there exists some $\beta \geq 0$ such that $|P - |VC'|| \leq \beta|W(\text{pb}) \cdot n + T(\text{pb}, G) - W(\text{pb}) \cdot P - k'|$. By the second property of Lemma 4, we have $|VC'| = \frac{W(\text{pb}) \cdot n + T(\text{pb}, G) - k'}{W(\text{pb})}$. Since $P \leq |VC'|$, we have $|P - |VC'|| = |VC'| - P = \frac{W(\text{pb}) \cdot n + T(\text{pb}, G) - k'}{W(\text{pb})} - P = \frac{1}{W(\text{pb})}(W(\text{pb}) \cdot n + T(\text{pb}, G) - W(\text{pb}) \cdot P - k')$.

So $\beta = 1$ is sufficient in both cases, since $W(\text{ECOMI}) = 4$ and $W(\text{ECONSI}) = 1$, which implies $\frac{1}{W(\text{pb})} \leq 1$.

Altogether, we then have $|\mathsf{opt}_{\text{MIN-VC}}(G) - |VC'|| \leq 1 \cdot |\mathsf{opt}_{\text{pb}}(G_1, G_2) - k'|$.

We proved that the reduction (R, S) is an *L-reduction*. This implies that for two genomes G_1 and G_2, both problems ECONSI and ECOMI are **APX**–hard even if $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$. Theorem 1 is proved. □

We extend in Corollary 1 our results for the *intermediate* and *maximum matching* models.

Corollary 1. ICOMI, MCOMI, ICONSI *and* MCONSI *are* **APX***–hard even when genomes* G_1 *and* G_2 *are such that* $\text{occ}(G_1) = 1$ *and* $\text{occ}(G_2) = 2$.

Proof. The *intermediate* and *maximum* matching models are identical to the *exemplar* model when one of the two genomes contains no duplicates. Hence, the **APX**–hardness result for ECOMI (resp. ECONSI) also holds for ICOMI and MCOMI (resp. ICONSI and MCONSI). □

3 EBD is APX–hard

Consider two genomes G_1 and G_2 with duplicates, and let EBD (resp. IBD, MBD) be the problem which consists in finding an exemplarization (resp. intermediate matching, maximum matching) $(G'_1, G'_2, \mathcal{M})$ of (G_1, G_2)that minimizes the number of breakpoints between G'_1 and G'_2.

EBD has been proved to be **NP**–complete even if $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$ [7]. Some inapproximability results also exist: in particular, it has been proved in [13] that, in the general case, EBD cannot be approximated within a factor $c \log n$, where $c > 0$ is a constant, and cannot be approximated within a factor 1.36 when $\text{occ}(G_1) = \text{occ}(G_2) = 2$. Moreover, for two balanced genomes G_1 and G_2 such that $k = \text{occ}(G_1) = \text{occ}(G_2)$, several approximation algorithms for MBD are given. These approximation algorithms admit respectively a ratio of 1.1037 when $k = 2$ [17], 4 when $k = 3$ [17] and $4k$ in the general case [19]. We can conclude from the above results that IBD and MBD problems are also **NP**–complete, since when one genome contains no duplicates, *exemplar*, *intermediate* and *maximum matching* models are equivalent.

In this section, we improve the above results by showing that the three problems EBD, IBD and MBD are **APX**–hard, even when genomes G_1 and G_2 are such that $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$. The main result is Theorem 2 below, which will be completed by Corollary 2 at the end of the section.

Theorem 2. EBD *is* **APX***–hard even when genomes* G_1 *and* G_2 *are such that* $\text{occ}(G_1) = 1$ *and* $\text{occ}(G_2) = 2$.

To prove Theorem 2, we use an *L-Reduction* from Min-Vertex-Cover-3 to EBD. Let $G = (V, E)$ be a cubic graph with $V = \{v_1 \dots v_n\}$ and $E = \{e_1 \dots e_m\}$. For each i, $1 \leq i \leq n$, let e_{f_i}, e_{g_i} and e_{h_i} be the three edges which are incident to v_i in G with $f_i < g_i < h_i$. Let R' be the polynomial transformation which associates to G the following genomes G_1 and G_2, where each gene has a positive sign:
$G_1 = a_0\ a_1\ b_1\ a_2\ b_2 \dots a_n\ b_n\ c_1\ d_1\ c_2\ d_2 \dots c_m\ d_m\ c_{m+1}$
$G_2 = a_0\ a_n\ d_{f_n}\ d_{g_n}\ d_{h_n}\ b_n \dots a_2\ d_{f_2}\ d_{g_2}\ d_{h_2}\ b_2\ a_1\ d_{f_1}\ d_{g_1}\ d_{h_1}\ b_1\ c_1\ c_2 \dots c_m\ c_{m+1}$
with :

- $a_0 = 0$, and for each i, $1 \leq i \leq n$, $a_i = i$ and $b_i = n + i$
- $c_{m+1} = 2n + m + 1$, and for each i, $1 \leq i \leq m$, $c_i = 2n + i$ and $d_i = 2n + m + 1 + i$

We remark that there is no duplication in G_1, so $\text{occ}(G_1) = 1$. In G_2, only the genes d_i, $1 \leq i \leq m$, are duplicated and occur twice. Thus $\text{occ}(G_2) = 2$.

Let G be a cubic graph and VC be a vertex cover of G. Let G_1 and G_2 be the genomes obtained by $R'(G)$. We define F' to be the polynomial transformation which associates to VC, G_1 and G_2 the exemplarization $F'(VC) = (G_1, G_2^E)$ of (G_1, G_2) as follows. For each i such that $v_i \notin VC$, we remove from G_2 the genes d_{f_i}, d_{g_i} and d_{h_i}. Then, for each j, $1 \leq j \leq m$ such that d_j still has two occurrences in G_2, we arbitrarily remove one of these occurrences in order to obtain the genome G_2^E. Hence, (G_1, G_2^E) is an exemplarization of (G_1, G_2).

Given a cubic graph G, we construct G_1 and G_2 by the transformation $R'(G)$. Given an exemplarization (G_1, G_2^E) of (G_1, G_2), let S' be the polynomial transformation which associates to (G_1, G_2^E) the set $VC = \{v_i | 1 \leq i \leq n, a_i \text{ and } b_i \text{ are not consecutive in } G_2^E\}$. We claim that VC is a vertex cover of G. Indeed, let e_p, $1 \leq p \leq m$, be an edge of G. Genome G_2^E contains one occurrence of gene d_p since G_2^E is an exemplarization of G_2. By construction, there exists i, $1 \leq i \leq n$, such

that d_p is in $G_2^E[a_i, b_i]$ and such that e_p is incident to v_i. The presence of d_p in $G_2^E[a_i, b_i]$ implies that vertex v_i belongs to VC. We can conclude that each edge of G is incident to at least one vertex of VC.

Lemmas 5 and 6 below are used to prove that (R', S') is an *L-Reduction* from the MIN-VERTEX-COVER-3 problem to the EBD problem. Let $G = (V, E)$ be a cubic graph with $V = \{v_1, v_2 \dots v_n\}$ and $E = \{e_1, e_2 \dots e_m\}$ and let us construct (G_1, G_2) by the transformation $R'(G)$.

Lemma 5. *Let VC be a vertex cover of G and (G_1, G_2^E) the exemplarization given by $F'(VC)$. Then $|VC| = k \Rightarrow B(G_1, G_2^E) \leq n + 2m + k + 1$, where $B(G_1, G_2^E)$ is the number of breakpoints between G_1 and G_2^E.*

Proof. Suppose $|VC| = k$. Let us list the breakpoints between genomes G_1 and G_2^E obtained by $F'(VC)$. The pairs (b_i, a_{i+1}), $1 \leq i \leq n-1$, and (b_n, c_1) induce one breakpoint each. For all i, $1 \leq i \leq m$, each pair of the form (c_i, d_i) (resp. (d_i, c_{i+1})) induces one breakpoint. For all i, $1 \leq i \leq n$, such that $v_i \in VC$, (a_i, b_i) induces at most one breakpoint. Finally, the pair (a_0, a_1) induces one breakpoint. Thus there are at most $n + 2m + k + 1$ breakpoints of (G_1, G_2^E). □

Lemma 6. *Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and VC' be the vertex cover of G obtained by $S'(G_1, G_2^E)$. We have $B(G_1, G_2^E) = k' \Rightarrow |VC'| = k' - n - 2m - 1$.*

Proof. Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and VC' be the vertex cover obtained by $S'(G_1, G_2^E)$. Suppose $B(G_1, G_2^E) = k'$. For any exemplarization (G_1, G_2^E) of (G_1, G_2), the following breakpoints always occur: the pair (a_0, a_1) ; for each i, $1 \leq i \leq m$, each pair (c_i, d_i) and (d_i, c_{i+1}) ; for each i, $1 \leq i \leq n-1$, the pair (b_i, a_{i+1}) ; the pair (b_n, c_1). Thus, we have at least $n + 2m + 1$ breakpoints. The other possible breakpoints are induced by pairs of the form of (a_i, b_i). Since we have $B(G_1, G_2^E) = k'$, there are exactly $k' - n - 2m - 1$ such breakpoints. By construction of VC', the cardinality of VC' is equal to the number of breakpoints induced by pairs of the form (a_i, b_i). So, we have: $|VC'| = k' - n - 2m - 1$. □

To prove that (R', S') is an *L-reduction*, we first notice that properties *(a)* and *(b)* of an *L-reduction* are trivially verified. The next lemma proves property *(c)*.

Lemma 7. *The inequality $\mathsf{opt}_{\text{EBD}}(G_1, G_2) \leq 12 \cdot \mathsf{opt}_{\text{MIN-VC}}(G)$ holds.*

Proof. For a cubic graph G with n vertices and m edges, we have $2m = 3n$ (see (4)) and $\mathsf{opt}_{\text{MIN-VC}}(G) \geq \frac{n}{2}$(see (5)). By construction of the genomes G_1 and G_2, any exemplarization of (G_1, G_2) contains $2n+2m+2$ genes in each genome. Thus, we have $\mathsf{opt}_{\text{EBD}}(G_1, G_2) \leq 2n+2m+2 \leq 6n$ ($n \geq 4$ in a cubic graph). Hence, we conclude that $\mathsf{opt}_{\text{EBD}}(G_1, G_2) \leq 12 \cdot \mathsf{opt}_{\text{MIN-VC}}(G)$. □

Now, we prove property (d) of our *L-reduction*.

Lemma 8. *Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and let VC' be the vertex cover of G obtained by $S'(G_1, G_2^E)$. Then, we have $|\mathsf{opt}_{\text{MIN-VC}}(G) - |VC'|| \leq |\mathsf{opt}_{\text{EBD}}(G_1, G_2) - B(G_1, G_2^E)|$*

Proof. Let (G_1, G_2^E) be an exemplarization of (G_1, G_2) and VC' be the vertex cover of G obtained by $S'(G_1, G_2^E)$. Let VC be a vertex cover of G such that $|VC| = \mathsf{opt}_{\text{MIN-VC}}(G)$.

We know that $\mathsf{opt}_{\text{MIN-VC}}(G) \leq |VC'|$ and $\mathsf{opt}_{\text{EBD}}(G_1, G_2) \leq B(G_1, G_2^E)$. So, it is sufficient to prove $|VC'| - \mathsf{opt}_{\text{MIN-VC}}(G) \leq B(G_1, G_2^E) - \mathsf{opt}_{\text{EBD}}(G_1, G_2)$.

By Lemma 5, we have $B(F'(VC)) \leq n + 2m + 1 + \mathsf{opt}_{\text{MIN-VC}}$, which implies $\mathsf{opt}_{\text{EBD}}(G_1, G_2) \leq B(F'(VC)) \leq n + 2m + 1 + \mathsf{opt}_{\text{MIN-VC}}$. Then

$$B(G_1, G_2^E) - \mathsf{opt}_{EBD}(G_1, G_2) \geq B(G_1, G_2^E) - n - 2m - 1 - \mathsf{opt}_{\text{MIN-VC}}(G) \tag{8}$$

By Lemma 6, we have: $|VC'| = B(G_1, G_2^E) - n - 2m - 1$ which implies

$$|VC'| - \mathsf{opt}_{\text{MIN-VC}}(G) = B(G_1, G_2^E) - n - 2m - 1 - \mathsf{opt}_{\text{MIN-VC}}(G) \tag{9}$$

Finally, by (8) and (9), we get $|VC'| - \mathsf{opt}_{\text{MIN-VC}} \leq B(G_1, G_2^E) - \mathsf{opt}_{\text{EBD}}(G_1, G_2)$. □

Lemmas 7 and 8 prove that the pair (R', S') is an *L-reduction* from MIN-VERTEX-COVER-3 to EBD. Hence, EBD is **APX**–hard even if $\text{occ}(G_1) = 1$ and $\text{occ}(G_2) = 2$, and Theorem 2 is proved. We extend in Corollary 2 our results for the *intermediate* and *maximum matching* models.

Corollary 2. *The* IBD *and* MBD *problems are* **APX***–hard even when genomes G_1 and G_2 are such that* $\text{occ}(G_1) = 1$ *and* $\text{occ}(G_2) = 2$.

Proof. The intermediate and maximum matching models are identical to the exemplar model when one of the two genomes contains no duplicates. Hence, the **APX**–hardness result for EBD also holds for IBD and MBD. □

4 Zero breakpoint distance

This section is devoted to zero breakpoint distance recognition issues. Indeed, in [13], the authors showed that deciding whether the exemplar breakpoint distance between any two genomes is zero or not is **NP**–complete even when no gene occurs more than three times in both genomes, *i.e.*, instances of type $(3, 3)$. This important result implies that the exemplar breakpoint distance problem does not admit any approximation in polynomial-time, unless $\mathbf{P} = \mathbf{NP}$. Following this line of research, we first complement the result of [13] by proving that deciding whether the exemplar breakpoint distance between any two genomes is zero or not is **NP**–complete, even when no gene is duplicated more than twice in one of the genomes (the maximum number of duplications is however unbounded in the other genome). This result is next extended to the intermediate matching model and we give a practical - but exponential - algorithm for deciding whether the exemplar breakpoint distance between any two genomes is zero or not in case no gene occurs more than twice in both genomes (a problem whose complexity, **P** *versus* **NP**–complete, remains open). Finally, we show that deciding whether the maximum matching breakpoint distance between any two genomes is zero or not is polynomial-time solvable and hence that such negative approximation results (the ones we obtained for the exemplar and intermediate models) do no propagate to the maximum matching model.

The following easy observation will prove extremely useful in the sequel of the present section.

Observation 3 *Let G_1 and G_2 be two genomes. If the exemplar breakpoint distance between G_1 and G_2 is zero, then there exists an exemplarization (G_1^E, G_2^E) of (G_1, G_2) such that (1) $G_1^E = G_2^E$, or (2) $-(G_1^E)^r = G_2^E$, where $-(G_1^E)^r$ is the signed reversal of genome G_1. The same observation can be made for the intermediate and maximum matching models.*

4.1 Zero exemplar breakpoint distance

The zero exemplar breakpoint distance (ZEBD) problem is formally defined as follows.

Problem: ZEBD
Input: Two genomes G_1 and G_2.
Question: Is the exemplar breakpoint distance between G_1 and G_2 equal to zero?

Aiming at precisely defining the inapproximability landscape of computing the exemplar breakpoint distance between two genomes, we complement the result of [13], who showed ZEBD to be **NP**–complete even for instances of type $(3, 3)$, by the following theorem.

Theorem 4. ZEBD *is* **NP***–complete even if no gene occurs more than twice in* G_1*.*

Proof. Membership of ZEBD to **NP** is immediate. The reduction we use to prove hardness is from Min-Vertex-Cover [16]. Let an arbitrary instance of Min-Vertex-Cover be given by a graph $G = (V, E)$ and a positive integer k. Write $V = \{v_1, v_2 \dots v_n\}$ and $E = \{e_1, e_2 \dots e_m\}$. In the rest of the proof, elements of V (resp. E) will be seen either as vertices (resp. edges) or genes, depending on the context. The corresponding instance (G_1, G_2) of ZEBD is defined as follows:

$$G_1 = v_1\ X_1\ v_2\ X_2 \dots v_n\ X_n$$
$$G_2 = Y[1]\ Y[2]\ \dots\ Y[k]\ Y_V.$$

For each $i = 1, 2, \dots, n$, X_i is defined to be $X_i = e_{i_1}\ e_{i_2}\ \dots\ e_{i_j}$, where $e_{i_1}, e_{i_2}, \dots, e_{i_j}$, $i_1 < i_2 < \dots < i_j$, are the edges incident to vertex v_i. The strings $Y[i]$, $1 \leq i \leq k$, are all equal and are defined by $Y[i] = Y_V\ Y_E$ where $Y_V = v_1\ v_2\ \dots\ v_n$ and $Y_E = e_1\ e_2\ \dots\ e_m$.

Notice that no gene occurs more than twice in G_1 (actually genes v_i occur once and genes e_i occur twice). However, the number of occurrences of each gene in G_2 is upper bounded by $k + 1$. Furthermore, all genes have positive sign, and hence according to Observation 3 we only need to consider exemplarizations (G_1^E, G_2^E) of (G_1, G_2) such that $G_1^E = G_2^E$.

It is immediate to check that our construction can be carried out in polynomial-time. We now claim that there exists a vertex cover of size k in G iff the exemplar breakpoint distance between G_1 and G_2 is zero.

Suppose first that there exists a vertex cover $V' \subseteq V$ of size k in G. Write $V' = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$, $i_1 < i_2 < \dots < i_k$. For convenience, we also define i_0 to be 0. From V' we construct an exemplarization (G_1^E, G_2^E) as follows. We obtain G_1^E from G_1 by a two step procedure. First we delete in G_1 all strings X_i such that $v_i \notin V'$. Second, for each $1 \leq j \leq m$, if gene e_j still occurs twice, we delete its second occurrence (this second step is concerned with edges connecting two vertices in V'). We now turn to G_2^E. For $1 \leq j \leq k$, we consider the string $Y[j] = Y_V\ Y_E$ that we process as follows: (1) we delete in Y_V all genes but v_{i_j} and those genes $v_\ell \notin V'$ such that $i_{j-1} < \ell < i_j$, and (2) we delete in Y_E genes e_ℓ that are not incident to v_{i_j}. Then, we also delete in Y_E genes e_ℓ that are incident to v_{i_j} and some smaller vertex in V' (*i.e.*, $e_\ell = \{v_{i_{j'}}, v_{i_j}\}$ for some $j' < j$). Finally, we delete in the trailing string $Y_V = v_1\ v_2\ \dots\ v_n$ all genes but those v_ℓ ($\notin V'$) such that $i_k < \ell$. Since V' is a vertex cover in G, then it follows that each gene occurs once in the obtained genomes, *i.e.*, (G_1^E, G_2^E) is indeed an exemplarization of (G_1, G_2). It is now easily seen that $G_1^E = G_2^E$, and hence that the exemplar breakpoint distance between G_1 and G_2 is zero.

Conversely, suppose that the exemplar breakpoint distance between G_1 and G_2 is zero. Since all genes have a positive sign, then it follows that there exists an exemplarization (G_1^E, G_2^E) of (G_1, G_2) such that $G_1^E = G_2^E$. Exemplarization G_2^E can be written as

$$G_2^E = Y_V[1]\ Y_E[1]\ Y_V[2]\ Y_E[2]\ \ldots Y_V[k]\ Y_E[k]\ Y_V[k+1]$$

where, $Y_V[i]$, $1 \leq i \leq k+1$, is a string on V and $Y_E[i]$, $1 \leq i \leq k$, is a string on E, V and E being viewed as alphabets. Now, define $V' \subseteq V$ as follows: $v_i \in V'$ iff gene v_i occurs in some $Y_V[j]$, $1 \leq j \leq k$, as the last gene. By construction, $|V'| \leq k$ (we may indeed have $|V'| < k$ if some $Y_V[j]$, $1 \leq j \leq k$, denotes the empty string). We now observe that, since no gene v_i is duplicated in G_1, all genes e_ℓ that occur between some gene $v_i \in V'$ and some gene $v_j \in V$ in G_2^E should match genes in string X_i in G_1. Then it follows that V' is a vertex cover of size at most k in G. □

The complexity of ZEBD remains open in case no gene occurs more than twice in G_1 and more than a constant times in G_2, *i.e.*, instances of type $(2, c)$ for some $c = O(1)$; recall here that ZEBD is **NP**–complete if no gene occurs more than three times in G_1 or in G_2 (instances of type $(3, 3)$, [13]). In particular, the complexity of ZEBD for instances of type $(2, 2)$ is open. However, we propose here a practical - but exponential - algorithm for ZEBD for instances of type $(2, 2)$, which is well-suited in case the number of genes that occur twice both in G_1 and in G_2 is relatively small.

Proposition 1. ZEBD *for instances of type* $(2, 2)$ *(no gene occurs more than twice in* G_1 *and in* G_2*) is solvable in* $O^*(1.6182^{2k})$ *time, where* k *is upper-bounded by the number of genes that occur exactly twice in* G_1 *and in* G_2*.*

Proof. According to Observation 3, for any instance (G_1, G_2), we only need to focus on exemplarizations (G_1^E, G_2^E) such that $G_1^E = G_2^E$ or $-(G_1^E)^r = G_2^E$, where $-(G_1^E)^r$ is the signed reversal of G_1^E. Let us first consider the case $G_1^E = G_2^E$ (the case $-(G_1^E)^r = G_2^E$ is identical up to a signed reversal and will thereby be briefly discussed at the end of the proof).

Let (G_1, G_2) be an instance of type $(2, 2)$ of ZEBD. Our algorithm is by transforming instance (G_1, G_2) into a CNF boolean formula ϕ with only few large clauses such that ϕ is satisfiable iff the exemplar breakpoint distance between G_1 and G_2 is zero. By hypothesis, each signed gene occurs at most twice in G_1 and in G_2. Therefore, for any signed gene g, we have one out of four possible distinct configurations depicted in Figure 2, where p_1, p_2, q_1 and q_2 are positions of occurrence of g in G_1 and G_2. Furthermore, since we are looking for an exemplarization (G_2^E, G_2^E) of (G_1, G_2) such that $G_1^E = G_2^E$, we may assume, in case g occurs only once in G_1 or in G_2, that all occurrences of g have the same sign (otherwise a trivial self-reduction would indeed apply). In other words, referring at Figure 2, we assume $G_1[p_1] = G_2[q_1] = G_2[q_2]$ in case (2), $G_1[p_1] = G_1[p_2] = G_2[q_1]$ in case (3), and $G_1[p_1] = G_2[q_1]$ in case (4). Finally, as for case (1), we may assume that either all occurrences have the same sign, or $G_1[p_1] = -G_1[p_2]$ and $G_2[q_1] = -G_2[q_2]$ (otherwise a trivial self-reduction would again apply).

We now describe the construction of the CNF boolean formula ϕ. First, the set of boolean variables X is defined as follows: for each gene g occurring at position p in G_1 and at position q in G_2 (*i.e.*, $|G_1[p]| = |G_2[q]|$) we add to X the boolean variable x_q^p. We now turn to defining the clauses of ϕ. Let g be any gene, and let the occurrence positions of g in G_1 and in G_2 be noted as in Figure 2.

– if $\mathrm{occ}(g, G_1) = \mathrm{occ}(g, G_2) = 2$ (case(1)),

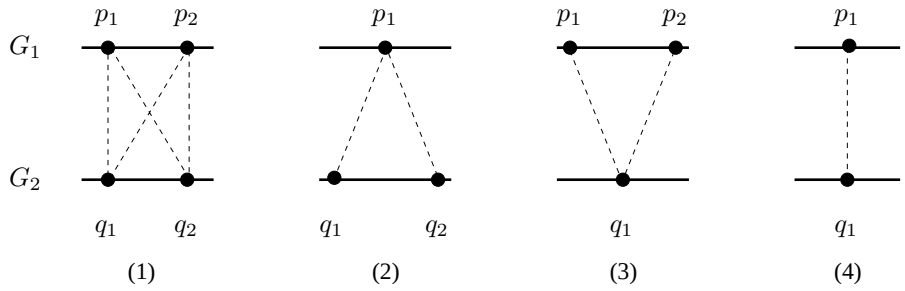

Fig. 2. The 4 gene-configurations for instances of type $(2, 2)$: p_1 and p_2 are the occurrence positions of gene g in G_1, and q_1 and q_2 are the occurrence positions of gene g in G_2.

- if $G_1[p_1] = G_1[p_2] = G_2[q_1] = G_2[q_2]$, we add to ϕ the clauses $(x_{q_1}^{p_1} \vee x_{q_2}^{p_1} \vee x_{q_1}^{p_2} \vee x_{q_2}^{p_2})$, $(\overline{x_{q_1}^{p_1}} \vee \overline{x_{q_2}^{p_1}})$, $(\overline{x_{q_1}^{p_1}} \vee \overline{x_{q_1}^{p_2}})$, $(\overline{x_{q_1}^{p_1}} \vee \overline{x_{q_2}^{p_2}})$, $(\overline{x_{q_2}^{p_1}} \vee \overline{x_{q_1}^{p_2}})$, $(\overline{x_{q_2}^{p_1}} \vee \overline{x_{q_2}^{p_2}})$ and $(\overline{x_{q_1}^{p_2}} \vee \overline{x_{q_2}^{p_2}})$,
- otherwise, we have $G_1[p_1] = -G_1[p_2]$ and $G_2[q_1] = -G_2[q_2]$ (see above discussion),
  - if $G_1[p_1] = G_2[q_1]$ and $G_1[p_2] = G_2[q_2]$)), we add to ϕ the clauses $(x_{q_1}^{p_1} \vee x_{q_2}^{p_2})$ and $(\overline{x_{q_1}^{p_1}} \vee \overline{x_{q_2}^{p_2}})$,
  - if $G_1[p_1] = G_2[q_2]$ and $G_1[p_2] = G_2[q_1]$)), we add to ϕ the clauses $(x_{q_2}^{p_1} \vee x_{q_1}^{p_2})$ and $(\overline{x_{q_2}^{p_1}} \vee \overline{x_{q_1}^{p_2}})$,

- if $\mathrm{occ}(g, G_1) = 1$ and $\mathrm{occ}(g, G_2) = 2$ (case (2)), we add to ϕ the clauses $(x_{q_1}^{p_1} \vee x_{q_2}^{p_1})$ and $(\overline{x_{q_1}^{p_1}} \vee \overline{x_{q_2}^{p_1}})$,
- if $\mathrm{occ}(g, G_1) = 2$ and $\mathrm{occ}(g, G_2) = 1$ (case (3)), we add to ϕ the clauses $(x_{q_1}^{p_1} \vee x_{q_1}^{p_2})$ and $(\overline{x_{q_1}^{p_1}} \vee \overline{x_{q_1}^{p_2}})$, and
- if $\mathrm{occ}(g, G_1) = \mathrm{occ}(g, G_2) = 1$ (case (4)), we add to ϕ the clause $(x_{q_1}^{p_1})$.

The rationale of this construction is that if formula ϕ evaluates to true for some assignment f and $f(x_q^p)$ is true for some gene g occurring at position p in G_1 and q in G_2, then all occurrences of g but the one at position p should be deleted in G_1 and all occurrences of g but the one at position q should be deleted in G_2, in order to obtain the exemplar solution. What is left is to enforce that ϕ evaluates to true iff the exemplar breakpoint distance between G_1 and G_2 is zero. To this aim, we add to ϕ the following clauses. For each pair of variables $(x_{j_1}^{i_1}, x_{j_2}^{i_2})$ such that $|G_1[i_1]| \neq |G_1[i_2]|$, $i_1 < i_2$ and $j_1 > j_2$, we add to ϕ the clause $(\overline{x_{j_1}^{i_1}} \vee \overline{x_{j_2}^{i_2}})$. The construction of ϕ is now complete.

Clearly, ϕ evaluates to true iff the exemplar breakpoint distance between G_1 and G_2 is zero. Let k be the number of genes g that occur twice in G_1 and in G_2 with the same sign, *i.e.*, $G_1[p_1] = G_1[p_2] = G_2[q_1] = G_2[q_2]$. We now make the important observation that all clauses in ϕ have size less than or equal to 2 except those k clauses of size 4 introduced in case gene g occurs twice in G_1 and in G_2 with the same sign. By introducing a new boolean variable, we can easily replace in ϕ each clause of size 4 by two clauses of size 3, and hence we may now assume that ϕ is a 3-CNF formula (*i.e.*, each clause has size at most 3) with exactly $2k$ clauses of size 3.

As for the case $-(G_1^E)^r = G_2^E$, we replace G_1 by $-(G_1)^r$ and construct another 3-CNF formula ϕ' as described above. The two 3-CNF formulas need, however, to be examined separately.

Fernau proposed in [15] an algorithm for solving 3-CNF boolean formulas that runs in $O^*(1.6182^\ell)$ time, where ℓ is the number of clauses of size 3. Therefore, ZEBD for instances of type $(2, 2)$ is solvable in $O^*(1.6182^{2k})$ time, where k is the number of genes g that occur twice in G_1 and in G_2. □

4.2 Zero intermediate matching breakpoint distance

We now turn to the zero intermediate breakpoint distance (ZIBD) problem. It is defined as follows.

Problem: ZIBD
Input: Two genomes G_1 and G_2.
Question: Is the intermediate breakpoint distance between G_1 and G_2 equal to zero ?

We show here that ZEBD and ZIBD are equivalent problems. We need the following lemma.

Lemma 9 ([2]). *Let G_1 and G_2 be two genomes without duplicates and with the same gene content, and G'_1 and G'_2 be the two genomes obtained from G_1 and G_2 by deleting any gene g. Then $B(G'_1, G'_2) \leq B(G_1, G_2)$.*

Theorem 5. ZEBD *and* ZIBD *are equivalent problems.*

Proof. One direction is trivial (any exemplarization is indeed an intermediate matching). The other direction follows from Lemma 9. □

It follows from Theorem 5 that the problem IBD is not approximable even for instances of type $(3, 3)$ (see [13]) and if no gene occurs more than twice in G_1 (see Theorem 4).

4.3 Zero maximum matching breakpoint distance

We show here that, oppositely to the exemplar and the intermediate matching models, deciding whether the maximum matching breakpoint distance between two genomes is equal to zero is polynomial-time solvable, and hence we cannot rule out the existence of accurate approximation algorithms for the maximum matching model. We refer to this problem as ZMBD.

Problem: ZMBD
Input: Two genomes G_1 and G_2.
Question: Is the maximum matching breakpoint distance between G_1 and G_2 equal to zero ?

The main idea of our approach is to transform any instance of ZMBD into a *matching diagram* and next use an efficient algorithm for finding a large set of non-intersecting line segments. Note that this latter problem is equivalent to finding a large increasing subsequence in permutations.

A matching diagram [18] consists of, say n, points on each of two parallel lines, and n straight line segments matching distinct pairs of points. The intersection graph of the line segments is called a *permutation graph* (the reason for the name is that if the points on the top line are numbered $1, 2, \ldots, n$, then the points on the other line are numbered by a permutation on $1, 2, \ldots, n$).

We describe how to turn the pair of genomes (G_1, G_2) into a matching diagram $D(G_1, G_2)$. For sake of presentation we introduce the following notation. For each gene family g, we write $\mathrm{occ}_{\mathrm{pos}}(G, g)$ (resp. $\mathrm{occ}_{\mathrm{neg}}(G, g)$) for the number of positive (resp. negative) occurrences of gene g in genome G. According to Observation 3, it is enough to consider two cases: $G_1^M = G_2^M$ or $-(G_1^M)^r = G_2^M$, where $(G_1^M, G_2^M, \mathcal{M})$ is a maximum matching of (G_1, G_2).

Let us first focus on testing $G_1^M = G_2^M$ (the case $-(G_1^M)^r = G_2^M$ is identical up to a signed reversal). We describe the construction of the top labeled points. Reading genome G_1 from left to right, we replace gene g by the sequence of labeled points

$$+\mathsf{g}_1(i, \mathrm{occ}_{\mathrm{pos}}(G_2, g)) \;\; +\mathsf{g}_1(i, \mathrm{occ}_{\mathrm{pos}}(G_2, g) - 1) \;\; \ldots \;\; +\mathsf{g}_1(i, 1)$$

if g is the i-th positive occurrence of gene g in genome G_1 or by the sequence of labeled points

$$-\mathtt{g}_1(i, \mathrm{occ}_{\mathrm{neg}}(G_2, g)) \ -\mathtt{g}_1(i, \mathrm{occ}_{\mathrm{neg}}(G_2, g) - 1) \ \ldots \ -\mathtt{g}_1(i, 1)$$

if g is the i-th negative occurrence of gene g in genome G_1. A symmetric construction is performed for the labeled points of the bottom line, *i.e.*, reading genome G_2 from left to right, we replace gene g by the sequence of labeled points

$$+\mathtt{g}_2(i, \mathrm{occ}_{\mathrm{pos}}(G_1, g)) \ +\mathtt{g}_2(i, \mathrm{occ}_{\mathrm{pos}}(G_1, g) - 1) \ \ldots \ +\mathtt{g}_2(i, 1)$$

if g is the i-th positive occurrence of gene g in genome G_2 or by the sequence of labeled points

$$-\mathtt{g}_2(i, \mathrm{occ}_{\mathrm{neg}}(G_1, g)) \ -\mathtt{g}_2(i, \mathrm{occ}_{\mathrm{neg}}(G_1, g) - 1) \ \ldots \ -\mathtt{g}_2(i, 1)$$

if g is the i-th negative occurrence of gene g in genome G_2. We now obtain the matching diagram $D(G_1, G_2)$ as follows: each labeled point $+\mathtt{g}_1(i, j)$ (resp. $-\mathtt{g}_1(i, j)$) of the top line is connected to the labeled point $+\mathtt{g}_2(j, i)$ (resp. $-\mathtt{g}_2(j, i)$) of the bottom line by a line segment. Clearly, each labeled point is incident to exactly one line segment, and hence $D(G_1, G_2)$ is indeed a matching diagram.

Of particular importance, observe that by construction, for any $x \in \{1, 2\}$ and any two labeled points $+\mathtt{g}_x(i, j)$ and $+\mathtt{g}_x(i, k)$, $j \neq k$, the two line segments incident to these two points are intersecting ; the same conclusion can be drawn for any two labeled points $-\mathtt{g}_x(i, j)$ and $-\mathtt{g}_x(i, k)$, $j \neq k$. The following lemma states this property in a suitable way.

Lemma 10. *If* $[+\mathtt{g}_1(i, j), +\mathtt{g}_2(j, i)]$ *and* $[+\mathtt{g}_1(k, \ell), +\mathtt{g}_2(\ell, k)]$ *(resp.* $[-\mathtt{g}_1(i, j), -\mathtt{g}_2(j, i)]$ *and* $[-\mathtt{g}_1(k, \ell), -\mathtt{g}_2(\ell, k)]$*) are two non-intersecting line segments in the matching diagram* $D(G_1, G_2)$*, then* $i \neq k$ *and* $j \neq \ell$*.*

Theorem 6. ZMBD *is polynomial-time solvable.*

Proof. Let G_1 and G_2 be two genomes, and m the size of a maximum matching between G_1 and G_2. According to Lemma 10, there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) such that $G_1^M = G_2^M$ if there exists m non-intersecting line segments in $D(G_1, G_2)$. The maximum number of non-intersecting line segments in a matching diagram with n points on each line can be found in $O(n \log \log n)$ time [8].

As for the case $-(G_1^M)^r = G_2^M$, we replace G_1 by $-(G_1)^r$ and run the same algorithm on the obtained matching diagram. □

5 Approximating the number of adjacencies in the maximum matching model

For two balanced genomes G_1 and G_2, several approximation algorithms for computing the number of *breakpoints* between G_1 and G_2 are given for the maximum matching model [17, 19]. We propose in this section three approximation algorithms to maximize the number of *adjacencies* (as opposed to minimizing the number of breakpoints). The approximation ratios we obtain are 1.1442 when $\mathrm{occ}(G_1) = 2$, $(3 + \epsilon)$ when $\mathrm{occ}(G_1) = 3$ (for any $\epsilon > 0$) and 4 in the general case. Observe that in the latter case, oppositely to [17, 19], our approximation ratio is independent of the maximum number of duplicates. Note also that in [12], inapproximation results are given for two *unbalanced* genomes G_1 and G_2 even when $\mathrm{occ}(G_1) = 1$ and $\mathrm{occ}(G_1) = 2$.

We first define the problem Max-k-Adj we are interested in ($k \geq 1$ is a fixed integer).

Problem: MAX-k-ADJ
Input: Two balanced genomes G_1 and G_2 with $\text{occ}(G_1) = k$ (and consequently $\text{occ}(G_2) = k$).
Solution: A maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2).
Measure: The number of adjacencies between G_1^M and G_2^M.

We define MAX-ADJ to be the problem MAX k-ADJ, in which k is unbounded.

5.1 A 1.1442-approximation for Max-2-Adj

We focus here on balanced genomes G_1 and G_2 such that $\text{occ}(G_1) = 2$, and we give an approximation algorithm for MAX-2-ADJ based on the MAX-2-CSP problem (defined below), for which a 1.1442-approximation algorithm is given in [9]. The main idea is to construct a boolean formula φ for each possible adjacency, and next to maximize the number of boolean formulas ϕ that can be simultaneously satisfied in a truth assignment ; the number of simultaneously satisfied formulas will be exactly the number of adjacencies, and hence any approximation ratio for MAX-2-CSP is an approximation ratio for MAX-2-ADJ.

Problem: MAX-k-CSP
Input: A pair (χ, Φ), where χ is a set of boolean variables and Φ is a set of boolean formulas such that each formula contains at most k literals of χ.
Solution: An assignment of χ.
Measure: The number of formulas that are satisfied by the assignment.

We define the following transformation MakeCSP that associates to any instance of MAX-2-ADJ an instance of MAX-2-CSP. Given an instance (G_1, G_2) of MAX-2-ADJ, we create a variable X_g for each gene g and define χ as the set of variables X_g. Then, we construct the set Φ of formulas. For each duo $d_i = (G_1[i], G_1[i+1])$, $1 \leq i \leq n_{G_1} - 1$, such that d_i or $-d_i$ appears in G_2, we distinguish three cases in order to create a formula φ_i of Φ:

1. There exists a unique duo $d_j = (G_2[j], G_2[j+1])$ in G_2 such that $d_j = d_i$ or $d_j = -d_i$. For sake of readability, we define the literal Y_p^q, $1 \leq p \leq n_{G_1}$, $1 \leq q \leq n_{G_2}$, where $|G_1[p]| = |G_2[q]|$, as follows: $Y_p^q = X_{|G_1[p]|}$ if $N_{G_1}[p] = N_{G_2}[q]$ and $Y_p^q = \overline{X_{|G_1[p]|}}$ otherwise. We now consider two cases:
   - (a) $d_i = d_j$: in that case, $\varphi_i = (Y_i^j \wedge Y_{i+1}^{j+1})$.
   - (b) $d_i = -d_j$: in that case, $\varphi_i = (Y_i^{j+1} \wedge Y_{i+1}^j)$.
2. The duo d_i appears twice in G_2. We consider two cases:
   - (c) $N_{G_1}[i] = N_{G_1}[i+1]$: in that case, $\varphi_i = (\overline{X_{|G_1[i]|} \oplus X_{|G_1[i+1]|}})$ where $\oplus$ is the boolean function XOR.
   - (d) $N_{G_1}[i] \neq N_{G_1}[i+1]$: in that case, $\varphi_i = (X_{|G_1[i]|} \oplus X_{|G_1[i+1]|})$.

Remark that each formula φ_i contains two literals. Hence, (χ, Φ) is an instance of MAX-2-CSP.

Lemma 11. *Let G_1 and G_2 be two balanced genomes such that $\text{occ}(G_1) = 2$. Let (χ, Φ) be the instance of* MAX-2-CSP *obtained by* MakeCSP(G_1, G_2). *For any integer k, if there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) which induces at least k adjacencies, then there exists an assignment of the variables of χ such that at least k formulas of Φ are satisfied.*

Proof. Let G_1 and G_2 be two balanced genomes such that $\text{occ}(G_1) = 2$ and let (χ, Φ) be the instance of Max-2-CSP obtained by $\mathsf{MakeCSP}(G_1, G_2)$. Let k be an integer.

Suppose there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) which induces at least k adjacencies. We construct the following assignment of variables of χ. For each gene g, we define $X_g = 1$ if g is not duplicated, else we define $X_g = 1$ iff the occurrences of g are matched in the reading order (see Figure 3). We now show that for each duo which induces an adjacency between G_1^M and G_2^M, there exists a distinct satisfied formula of Φ. Let $d_i = (G_1^M[i], G_1^M[i+1])$, $1 \leq i \leq n_{G_1} - 1$, be a duo which induces an adjacency, and let $d_j = (G_2^M[j], G_2^M[j+1])$ be the related duo on G_2^M. By construction of Φ, there exists a formula $\varphi_i \in \Phi$ which has been previously defined in one of the cases (a), (b), (c) or (d) of the definition of $\mathsf{MakeCSP}$. We claim that, for each of these cases, φ_i is satisfied:

- (a) $\varphi_i = (Y_i^j \wedge Y_{i+1}^{j+1})$ and $d_i = d_j$. We first prove that literal Y_i^j is true. Three cases are possible. **(i)** The gene $|G_1[i]|$ is not duplicated ; then we have defined in our assignment $X_{|G_1[i]|} = 1$. Moreover, we have $Y_i^j = X_{|G_1[i]|}$ (since $N_{G_1}[i] = N_{G_2}[j] = 0$), hence Y_i^j is true. **(ii)** The gene $|G_1[i]|$ is duplicated and $N_{G_1}[i] = N_{G_2}[j]$; then, by definition of our assignment and since $G_1[i]$ and $G_2[j]$ are matched together in the maximum matching $(G_1^M, G_2^M, \mathcal{M})$, we have $X_{|G_1[i]|} = 1$ (we match signed genes in the reading order). Moreover, we have $Y_i^j = X_{|G_1[i]|}$ which induces that Y_i^j is true. **(iii)** The gene $|G_1[i]|$ is duplicated and $N_{G_1}[i] \neq N_{G_2}[j]$; then, by definition of our assignment and since $G_1[i]$ and $G_2[j]$ are matched together in the maximum matching $(G_1^M, G_2^M, \mathcal{M})$, we have $X_{|G_1[i]|} = 0$ (we do not match signed genes in the reading order). Moreover, we have in this case $Y_i^j = \overline{X_{|G_1[i]|}}$ which induces that Y_i^j is true.

  In each case, we have proved that Y_i^j is true. We can also prove that Y_{i+1}^{j+1} is true, using the same arguments. Hence, we conclude that φ_i is true.
- (b) $\varphi_i = Y_i^{j+1} \wedge Y_{i+1}^j$ and $d_i = -d_j$. By similar arguments as in case (a), we can prove that Y_i^{j+1} and Y_{i+1}^j are true.
- (c) We have $N_{G_1}[i] = N_{G_1}[i+1]$ and the duo d_i appears twice in G_2 (noted by d_j and $d_{j'}$). Since d_i induces an adjacency, the duo d_i matches either d_j or $d_{j'}$. In these two cases, we have $X_{|G_1[i]|} = X_{|G_1[i+1]|}$ (otherwise $G_1[i]$ and $G_1[i+1]$ would not match successive signed genes). Moreover, $\varphi_i = (\overline{X_{|G_1[i]|} \oplus X_{|G_1[i+1]|}})$ and thus, φ_i is true.
- (d) We have $N_{G_1}[i] \neq N_{G_1}[i+1]$ and the duo d_i appears twice in G_2 (noted by d_j and $d_{j'}$). Since d_i induces an adjacency, the duo d_i matches either d_j or $d_{j'}$. In these two cases, we have $X_{|G_1[i]|} \neq X_{|G_1[i+1]|}$ (otherwise $G_1[i]$ and $G_1[i+1]$ would not match successive signed genes). Moreover, $\varphi_i = (X_{|G_1[i]|} \oplus X_{|G_1[i+1]|})$ and thus, φ_i is true.

We have constructed a variable assignment of χ such that, for each duo d_i in G_1^M which implies an adjacency, there exists a distinct satisfied formula $\varphi_i \in \Phi$. Thus, if there exists a maximum matching of (G_1, G_2) which induces at least k adjacencies, then the corresponding assignment implies at least k satisfied formulas. □

Lemma 12. *Let G_1 and G_2 be two balanced genomes such that* $\text{occ}(G_1) = 2$. *Let (χ, Φ) be the instance of* Max-2-CSP *obtained by* $\mathsf{MakeCSP}(G_1, G_2)$. *For any integer k, if there exists an assignment of χ such that at least k formulas of Φ are satisfied, then there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) which induces at least k adjacencies.*

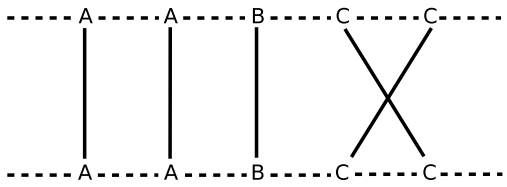

Fig. 3. All possibilities of assignment: $X_A = 1$ (gene A occurs twice and signed genes are matched in the reading order), $X_B = 1$ or $X_B = 0$ (gene B occurs once) and $X_C = 0$ (gene C occurs twice and signed genes are not matched in the reading order). Note that this construction is independent of the sign of the genes.

Proof. Let G_1 and G_2 be two balanced genomes such that $\text{occ}(G_1) = 2$ and let (χ, Φ) be the instance of Max-2-CSP obtained by $\mathsf{MakeCSP}(G_1, G_2)$. Let k be an integer.

Suppose there exists an assignment of χ such that at least k formulas $\varphi_i \in \Phi$ are satisfied. We create the following maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2). For each variable X_g such that the gene g is duplicated, we match the occurrences of g in the reading order if $X_g = 1$ (such as gene A in Figure 3). If we have $X_g = 0$, we match the first occurrence of g on G_1 with the second one on G_2 and the second occurrence of g on G_1 with the first one on G_2 (such as gene C in Figure 3). Then, we match signed genes which are not duplicated. Now, we prove that each satisfied formula $\varphi_i \in \Phi$ induces a distinct adjacency for $(G_1^M, G_2^M, \mathcal{M})$. Let $\varphi_i \in \Phi$ be a satisfied formula which is defined in one of the cases (a), (b), (c) or (d) of the definition of $\mathsf{MakeCSP}$:

- (a) We have $\varphi_i = (Y_i^j \wedge Y_{i+1}^{j+1})$ and the duos $d_i = (G_1[i], G_1[i+1])$ and $d_j = (G_2[j], G_2[j+1])$ are identical.
  Here, we must prove that d_i and d_j are matched together in $(G_1^M, G_2^M, \mathcal{M})$ and thus induce an adjacency. First, we show that signed genes $G_1[i]$ and $G_2[j]$ are matched together in $(G_1^M, G_2^M, \mathcal{M})$. Since φ_i is satisfied, we have $Y_i^j = 1$. We must dissociate three cases: **(i)** the gene $|G_1[i]|$ is not duplicated: in that case, the signed gene $G_1[i]$ can be matched only with $G_2[j]$. **(ii)** The gene $|G_1[i]|$ is duplicated and we have $N_{G_1}[i] = N_{G_2}[j]$. In that case, we have defined $Y_i^j = X_{|G_1[i]|}$ which implies $X_{|G_1[i]|} = 1$. Thus, since $N_{G_1}[i] = N_{G_2}[j]$, the signed genes $G_1[i]$ and $G_2[j]$ are matched together. **(iii)** The gene $|G_1[i]|$ is duplicated and we have $N_{G_1}[i] \neq N_{G_2}[j]$. In that case, we have defined $Y_i^j = \overline{X_{|G_1[i]|}}$ which implies $X_{|G_1[i]|} = 0$. Thus, since $N_{G_1}[i] \neq N_{G_2}[j]$, the signed genes $G_1[i]$ and $G_2[j]$ are matched together. For each case, the signed genes $G_1[i]$ and $G_2[j]$ are matched together. We can conclude in the same way that $G_1[i+1]$ and $G_2[j+1]$ are also matched together, which implies that d_i induces an adjacency.
- (b) We have $\varphi_i = (Y_i^{j+1} \wedge Y_{i+1}^j) = 1$ and the duos $d_i = (G_1[i], G_1[i+1])$ and $d_j = (G_2[j], G_2[j+1])$ are reversed.
  We can use the same reasoning used in case (a) to prove that d_i induces an adjacency.
- (c) The duo d_i appears twice in G_2 (noted by d_j and $d_{j'}$). We have $\varphi_i = (\overline{X_{|G_1[i]|} \oplus X_{|G_1[i+1]|}})$ and $N_{G_1}[i] = N_{G_1}[i+1]$.
  Since φ_i is true, we have $X_{|G_1[i]|} = X_{|G_1[i+1]|}$ which implies by construction of the maximum matching that d_i matches d_j or $d_{j'}$.
- (d) The duo d_i appears twice in G_2 (noted by d_j and $d_{j'}$). We have $\varphi_i = (X_{|G_1[i]|} \oplus X_{|G_1[i+1]|})$ and $N_{G_1}[i] \neq N_{G_1}[i+1]$. Since φ_i is true, we have $X_{|G_1[i]|} \neq X_{|G_1[i+1]|}$ which implies by construction of the maximum matching that d_i matches d_j or $d_{j'}$.

Consequently, for each satisfied formula, there exists a distinct adjacency between G_1^M and G_2^M. Thus, if there exists an assignment of χ which implies at least k satisfied formulas of Φ, then there exists a maximum matching of (G_1, G_2) which implies at least k adjacencies. □

Lemmas 11 and 12 prove that any α-approximation for Max-2-CSP implies an α-approximation for Max-2-Adj. In [9], an approximation algorithm is given for Max-2-CSP, whose approximation ratio is equal to $\frac{1}{0.874} \leq 1.1442$. Thus, we have the following theorem.

Theorem 7. Max-2-Adj *is* 1.1442*-approximable.*

5.2 A $(3+\epsilon)$-approximation for Max-3-Adj

Now, we present a $(3+\epsilon)$-approximation for Max-3-Adj by using the Maximum Independent Set problem defined as follows:

Problem: Max-Independent-Set
Input: A graph $G = (V, E)$.
Solution: An independent set of G (i.e. a subset V' of V such that no two vertices in V' are joined by an edge in E).
Measure: The cardinality of V'.

In [17], Goldstein *et al.* used Max-Independent-Set to approximate the Minimum Common String Partition problem by creating a *conflict graph*. We construct in the same way an instance of Max-Independent-Set where a vertex represents a possible adjacency and where an edge represents a conflict between two adjacencies. We define MakeMIS to be the following transformation which associates to two balanced genomes G_1 and G_2 an instance of Max-Independent-Set. We construct a vertex for each duo match, and then we create an edge between two vertices when they are in conflict, i.e. when two matches are incompatible. Figure 4 illustrates the graph obtained by $\mathsf{MakeMIS}(G_1, G_2)$ where $G_1 = +3+1+2+3+4+2+5$ and $G_2 = +3+4+2+3+1+2+5$.

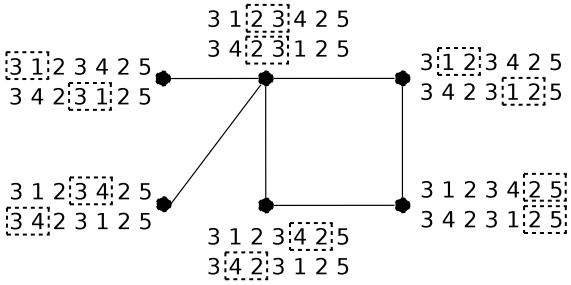

Fig. 4. The conflict graph obtained by $\mathsf{MakeMIS}(G_1, G_2)$ where $G_1 = +3+1+2+3+4+2+5$ and $G_2 = +3+4+2+3+1+2+5$ (for sake of readability, positive signs are not displayed).

In order to prove that there exists a $(3+\epsilon)$-approximation for Max-3-Adj, we give the following intermediate lemmas.

Lemma 13. *Let G_1 and G_2 be two balanced genomes and let G be the graph obtained by* MakeMIS(G_1, G_2). *For any integer k, there exists an independent set V' of G such that $|V'| \geq k$ iff there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) which induces at least k adjacencies.*

Proof. Let G_1 and G_2 be two balanced genomes and let G be the graph obtained by MakeMIS(G_1, G_2). Let k be an integer.

($\Rightarrow$) Suppose there exists an independent set V' of G such that $|V'| \geq k$. We construct a matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) as follows: first, for each vertex of V', we match together the two corresponding duos, thus inducing one adjacency (called a *definite* adjacency). By construction of G, this operation is possible. Indeed, two vertices which are not connected in G imply two compatible adjacencies. Then, we match arbitrarily the unmatched genes. This operation cannot break any definite adjacency. Finally, we obtain a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ which induces at least $|V'|$ adjacencies, and consequently at least k adjacencies.

($\Leftarrow$) Suppose there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) which induces at least k adjacencies. We construct a set V' by taking each vertex which represents a duo match between G_1^M and G_2^M. By construction of G, V' is an independent set (no pair of adjacencies can create a conflict), and then we have $|V'| \geq k$. □

Lemma 14. *Let G_1 and G_2 be two balanced genomes such that* $\mathrm{occ}(G_1) = k$. *The maximum degree Δ of the graph G obtained by* MakeMIS(G_1, G_2) *satisfies $\Delta \leq 6(k-1)$.*

Proof. Let G_1 and G_2 be two balanced genomes such that $\mathrm{occ}(G_1) = k$ and let G be the graph obtained by MakeMIS(G_1, G_2). Consider a duo match $m = (d_1, d_2)$ with $d_1 = (G_1[i], G_1[i+1])$ and $d_2 = (G_2[j], G_2[j+1])$ where $1 \leq i \leq n_{G_1} - 1$ and $1 \leq j \leq n_{G_2} - 1$. We claim that the vertex v_m of G, which represents the duo match m, is connected to at most $6(k-1)$ vertices. For this, we list the possible duo matches $m' = (d'_1, d'_2)$ such that the vertex $v_{m'}$ of G which represents m' is connected to v_m. Remark that if $v_{m'}$ is connected to v_m (i.e. m and m' are in conflict), then at least one of the duos d'_1 and d'_2 overlaps, respectively, either d_1 or d_2. Let d'_1 be a duo in G_1 which overlaps d_1. First, we list the possible duos d'_2 such that the duo matches $m = (d_1, d_2)$ and $m' = (d'_1, d'_2)$ are in conflict. Remark that d'_1 (or $-d'_1$) appears at most k times on G_2 since a gene can occur at most k times. We then distinguish three cases:

- (a) $d'_1 = (G_1[i-1], G_1[i])$: if d'_1 (or $-d'_1$) appears k times in G_2, one of these occurrences is necessary $d'_2 = (G_2[j-1], G_2[j])$ if $d_1 = d_2$, or $d'_2 = (G_2[j+1], G_2[j+2])$ if $d_1 = -d_2$. For these two cases, the duo matches m and (d'_1, d'_2) are not in conflict.
- (b) $d'_1 = d_1$: if d'_1 (or $-d'_1$) appears k times on G_2, one of these occurrences is necessary d_2, which induces in this case no conflict with m.
- (c) $d'_1 = (G_1[i+1], G_1[i+2])$: if d'_1 (or $-d'_1$) appears k times on G_2, one of these occurrences is necessary $d'_2 = (G_2[j+1], G_2[j+2])$ if $d_1 = d_2$, or $d'_2 = (G_2[j-1], G_2[j])$ if $d_1 = -d_2$. For these two cases, the duo matches m and m' are not in conflict.

For each case, one of the k possible duos d'_2 does not imply a conflict between m and m'. Thus, for any duo d'_1 which overlaps d_1, there exists at most $k-1$ duos d'_2 on G_2 such that m and m' are in conflict. Using the same arguments, we can easily prove that for any duo d'_2 which overlaps d_2, there exists at most $k-1$ duos d'_1 on G_1 such that m and m' are in conflict. Hence, each of the six duos which overlaps d_1 or d_2 implies at most $k-1$ conflicts. Thus, we obtain at most $6(k-1)$ vertices which are connected to the vertex v_m in the conflict graph. □

According to Lemma 13, any α-approximation for Max-Independent-Set is thus also an α-approximation for Max-k-Adj. In [5], Berman and Fürer present a polynomial time algorithm that approximates Max-Independent-Set within ratio depending of the degree Δ of the graph. For every $\Delta > 2$ and $\epsilon > 0$, the approximation ratio is $\frac{\Delta+3}{5} + \epsilon$ for even Δ, and $\frac{\Delta+3.25}{5} + \epsilon$ for odd Δ. Combining this with Lemma 14, we obtain the following result.

Theorem 8. *For every $\epsilon > 0$,* Max-k-Adj *is $(\frac{6k-3}{5} + \epsilon)$-approximable.*

Note that in the case where $k = 2$, we obtain a ratio of $1.8 + \epsilon$, which is not better than the one obtained in Theorem 7. Moreover, we introduce in the next section a 4-approximation in the general case. Hence, the only interesting case of Theorem 8 above is when $k = 3$, inducing a $(3 + \epsilon)$-approximation for Max-3-Adj.

5.3 A 4-approximation for Max-Adj

In [14], a 4-approximation algorithm for the Max-Weighted 2-interval Pattern problem (Max-W2IP) is given. In the following, we first define Max-W2IP, and next we present how we can relate any instance of Max-Adj to an instance of Max-W2IP.

The Maximum Weighted 2-Interval Pattern problem. A 2-*interval* is the union of two disjoint intervals defined over a single line. For a 2-interval $D = (I, J)$, we always assume that the interval $I < J$, *i.e.*, I is completely on the left of J does not overlap J. We say that two 2-intervals $D_1 = (I_1, J_1)$ and $D_2 = (I_2, J_2)$ are *disjoint* if D_1 and D_2 have no common point (i.e. $(I_1 \cup J_1) \cap (I_2 \cup J_2) = \emptyset$). Three possible relations exist between two disjoint 2-intervals: we write (1) $D_1 \prec D_2$, if $I_1 < J_1 < I_2 < J_2$, (2) $D_1 \sqsubset D_2$, if $I_2 < I_1 < J_1 < J_2$ and (3) $D_1 \between D_2$, if $I_1 < I_2 < J_1 < J_2$.

We say that a pair of 2-intervals D_1 and D_2 is *R-comparable* for some $R \in \{\prec, \sqsubset, \between\}$, if either $(D_1, D_2) \in R$ or $(D_2, D_1) \in R$. A set of 2-intervals $\mathcal{D}$ is $\mathcal{R}$-comparable for some $\mathcal{R} \subseteq \{\prec, \sqsubset, \between\}$, $\mathcal{R} \neq \emptyset$, if any pair of distinct 2-intervals in $\mathcal{D}$ is R-comparable for some $R \in \mathcal{R}$. The non-empty set $\mathcal{R}$ is called a *model*. The Max-Weighted 2-interval Pattern (Max-W2IP) problem is formally defined as follows.

Problem: Max-Weighted 2-interval Pattern (Max-W2IP)
Input: A set $\mathcal{D}$ of 2-intervals, a model $\mathcal{R} \subseteq \{\prec, \sqsubset, \between\}$ with $\mathcal{R} \neq \emptyset$, and a weight function $\omega : \mathcal{D} \to \mathbb{N}^+$.
Solution: An $\mathcal{R}$-comparable subset D' of $\mathcal{D}$.
Measure: The weight of D'.

Transformation. We first describe how to transform any instance (G_1, G_2) of Max-Adj into an instance, referred hereafter as $\mathsf{Make2I}(G_1, G_2) = (\mathcal{D}, \mathcal{R}, \omega)$, of Max-W2IP. We need a new definition. Let G_1 and G_2 be two balanced genomes. An interval I_1 of G_1 and an interval I_2 of G_2, both of size at least 2, are said to be *identical* if they correspond to the same string up to a complete reversal, where a reversal also changes all the signs in the string. Clearly, two identical intervals have the same length.

The weighted 2-interval set $\mathcal{D}$ is obtained as follows. We first concatenate G_1 and G_2, and for any pair (I_1, I_2) of identical intervals (I_1 is an interval of G_1 and I_2 is an interval of G_2), we construct the 2-interval $D = (I_1, I_2)$ of weight $\omega(D) = |I_1| - 1$ $(= |I_2| - 1)$ and add it to $\mathcal{D}$. Notice that, since identical intervals have length at least 2, each 2-interval of $\mathcal{D}$ has weight at least 1.

Figure 5 gives an example of such a construction. Observe that, by construction, no two 2-intervals of $\mathcal{D}$ are $\{\prec\}$-comparable. The construction of the instance of Max-W2IP is complete by setting $\mathcal{R} = \{\prec, \sqsubset, \between\}$, *i.e.*, we are looking for disjoint 2-intervals, no matter what the relation between any two disjoint 2-interval is. Therefore, for sake of abbreviation, we shall denote the corresponding instance simply as $\mathsf{Make2I}(G_1, G_2) = (\mathcal{D}, \omega)$ and forget about the model.

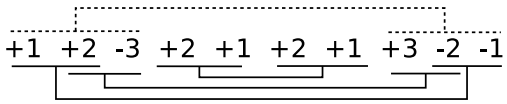

Fig. 5. 2-intervals induced by genomes $G_1 = +1\ +2\ -3\ +2\ +1$ and $G_2 = +2\ +1\ +3\ -2\ -1$. For readability, singleton intervals are not drawn. The dotted 2-interval is of weight 2, while all other 2-intervals are of weight 1.

We now describe how to transform any solution of Max-W2IP into a solution of Max-Adj. Let G_1 and G_2 be two balanced genomes and $\mathsf{Make2I}(G_1, G_2) = (\mathcal{D}, \omega)$. Furthermore, let $\mathcal{S} \subseteq \mathcal{D}$ be a set of disjoint 2-intervals, *i.e.* a solution for Max-W2IP for model the $\{\prec, \sqsubset, \between\}$ for the instance $(\mathcal{D}, \omega)$.

We write $\mathsf{Max\text{-}W2IP\_to\_Adj}(\mathcal{S})$ for the transformation of $\mathcal{S}$ into a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) defined as follows. First, for each 2-interval $D = (I_1, I_2)$ of $\mathcal{S}$, we match the signed genes of I_1 and I_2 in the natural way ; then, in order to achieve a maximum matching (since each signed gene is not necessarily covered by a 2-interval in $\mathcal{S}$), we apply the following greedy algorithm: iteratively, we match, arbitrarily, two unmatched signed genes g_1 and g_2 such that $|g_1| = |g_2|$ and g_i is a gene of G_i $(i = 1, 2)$, until no such pair of signed genes exists. After a relabeling of signed genes according to this matching (denoted $\mathcal{M}$), we obtain a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2).

The rationale of this construction stems from two following lemmas.

Lemma 15. *Let G_1 and G_2 be two balanced genomes, $\mathsf{Make2I}(G_1, G_2) = (\mathcal{D}, \omega)$ and $\mathcal{S}$ be any set of disjoint 2-intervals of $\mathcal{D}$. If we denote by $W_\mathcal{S}$ the total weight of $\mathcal{S}$, then the maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) obtained by $\mathsf{Max\text{-}W2IP\_to\_Adj}(\mathcal{S})$ induces at least $W_\mathcal{S}$ adjacencies.*

Proof. For each 2-interval $D = (I_1, I_2)$ of $\mathcal{S}$, we have matched the signed genes of I_1 and I_2 in the natural way. Therefore, for each 2-interval $D = (I_1, I_2)$ of $\mathcal{S}$, we obtain $|I_1| - 1$ adjacencies in $(G_1^M, G_2^M, \mathcal{M})$ since I_1 and I_2 are identical intervals. Since the final greedy part of $\mathsf{Max\text{-}W2IP\_to\_Adj}(\mathcal{S})$ does not delete any adjacency, we have at least $W_\mathcal{S}$ adjacencies in $(G_1^M, G_2^M, \mathcal{M})$. □

Lemma 16. *Let G_1 and G_2 be two balanced genomes, $(G_1^M, G_2^M, \mathcal{M})$ be a maximum matching of (G_1, G_2), $\mathsf{Make2I}(G_1, G_2) = (\mathcal{D}, \omega)$ and W be the number of adjacencies induced by $(G_1^M, G_2^M, \mathcal{M})$. Then there exists a subset $\mathcal{S} \subseteq \mathcal{D}$ of disjoint 2-intervals of total weight W.*

Proof. Denote by n the size of G_1^M. Consider any factorization $G_1^M = s_1\, s_2\, \ldots\, s_p$ such that, for each $1 \leq i < p$, s_i and s_{i+1} are separated by one breakpoint and no breakpoint appears in s_i, $1 \leq i \leq p$. Therefore, there exists $p-1$ breakpoints between G_1^M and G_2^M, and hence $n-p$ adjacencies between G_1^M and G_2^M . To each substring s_i of the factorization of G_1^M corresponds a substring t_i in G_2^M such that s_i and t_i are identical. Moreover, each substring s_i of size l_i, $1 \leq i \leq p$, contains $l_i - 1$

adjacencies. We construct the 2-interval set $\mathcal{S}$ as the union of $D_i = (\hat{s_i}, \hat{t_i})$, $1 \leq i \leq p$, where $\hat{s_i}$ (resp. $\hat{t_i}$) is the interval obtained from s_i (resp. t_i). The factorization of G_1^M implies that the constructed 2-intervals are disjoint, and hence the total weight of $\mathcal{S}$ is $\sum_{i=1}^{p}(l_i - 1) = \sum_{i=1}^{p} l_i - \sum_{i=1}^{p} 1 = n - p = W$. □

We now describe Algorithm ApproxAdj and then prove it to be a 4-approximation algorithm for Max-Adj.

Algorithm 1 ApproxAdj

Require: Two balanced genomes G_1 and G_2.
Ensure: A maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2).
- Let Make2I$(G_1, G_2) = (\mathcal{D}, \omega)$.
- Invoke the 4-approximation algorithm of Crochemore *et al.* [14] to obtain a set of disjoint 2-intervals $\mathcal{S} \subseteq \mathcal{D}$.
- Construct the maximal matching $(G_1^M, G_2^M, \mathcal{M}) =$ Max-W2IP_to_Adj$(\mathcal{S})$.

Theorem 9. *Algorithm* ApproxAdj *is a* 4*-approximation algorithm for* Max-Adj.

Proof. According to Lemmas 15 and 16, there exists a maximum matching $(G_1^M, G_2^M, \mathcal{M})$ of (G_1, G_2) that induces W adjacencies iff there exists a subset of disjoint 2-intervals $\mathcal{S} \subseteq \mathcal{D}$ with total weight W. Therefore, any approximation ratio for Max-W2IP implies the same approximation ratio for Max-Adj. In [14], a 4-approximation algorithm is proposed for Max-W2IP. Hence, Algorithm ApproxAdj is a 4-approximation algorithm for Max-Adj. □

6 Conclusions and future work

In this paper, we have first given new approximation complexity results for several optimization problems in genomic rearrangement. We focused on conserved intervals, common intervals and breakpoints, and we took into account the presence of duplicates. We restricted our proofs to cases where one genome contains no duplicates and the other contains no more than two occurrences of each gene. With this assumption, we proved that the problems consisting in computing an exemplarization (resp. an intermediate matching, a maximum matching) optimizing any of the three above mentioned measures is **APX**–hard, thus extending the results of [7, 10, 13]. In a second part of the paper, we have focused on the ZEBD (resp. ZIBD, ZMBD) problems, where the question is whether there exists an exemplarization (resp. intermediate matching, maximum matching) that induces zero breakpoint. We have extended a result from [13] by showing that ZEBD is **NP**–complete even for instances of type $(2, k)$, where k is unbounded. We also have noted that ZEBD and ZIBD are equivalent problems, and shown that ZMBD is in **P**. Finally, we gave several approximation algorithms for computing the maximum number of adjacencies of two balanced genomes under the *maximum matching* model. The approximation ratios we get are 1.1442 for instances of type $(2, 2)$, $(3 + \epsilon)$ for instances of type $(3, 3)$ with $\epsilon > 0$ and 4 in the general case. Concerning the latter result, we note that the approximation ratio we obtain is constant, even when the number of occurrences in genomes is unbounded.

However, several problems remain unsolved. In particular, concerning approximation algorithms, virtually nothing is known (i) in the case of unbalanced genomes and (ii) in the exemplar and intermediate models. Indeed, all the existing results (see for instance [17, 19] for the number of

breakpoints), including ours, focus on the maximum matching problem for balanced genomes, which implies that no gene is deleted from genomes G_1 and G_2. Now, if we allow genes to be deleted, the problem seems much more difficult to tackle.

Finally, we would like to recall the following open problem from [11]: what is the complexity of ZEBD for instances of type $(2, 2)$?

References

1. P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1-2):123–134, 2000.
2. S. Angibaud, G. Fertin, I. Rusu, A. Thévenin, and S. Vialette. Efficient tools for computing the number of breakpoints and the number of adjacencies between two genomes with duplicate genes. *Journal of Computational Biology*, 2008. Submitted.
3. S. Angibaud, G. Fertin, I. Rusu, and S. Vialette. A general framework for computing rearrangement distances between genomes with duplicates. *Journal of Computational Biology*, 14(4):379–393, 2007.
4. V. Bafna and P. Pevzner. Sorting by reversals: genome rearrangements in plant organelles and evolutionary history of X chromosome. *Molecular Biology and Evolution*, pages 239–246, 1995.
5. P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *Proc. SODA '94: Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 365–371, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.
6. G. Blin and R. Rizzi. Conserved interval distance computation between non-trivial genomes. In *Proc. COCOON 2005*, volume 3595 of *LNCS*, pages 22–31. Springer, 2005.
7. D. Bryant. The complexity of calculating exemplar distances. In *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignement, and the Evolution of Gene Families*, pages 207–212. Kluwer Academic Publisher, 2000.
8. M.-S. Chang and F.-H. Wang. Efficient algorithms for the maximum weight clique and maximum weight independent set problems on permutation graphs. *Information Processing Letters*, 43(6):293–295, 1992.
9. M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 62–68, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.
10. C. Chauve, G. Fertin, R. Rizzi, and S. Vialette. Genomes containing duplicates are hard to compare. In *Proc. IWBRA 2006*, volume 3992 of *LNCS*, pages 783–790. Springer, 2006.
11. Z. Chen, R. Fowler, B. Fu, and B. Zhu. Lower bounds on the approximation of the exemplar conserved interval distance problem of genomes. In *Proc COCOON 2006*, volume 4112 of *LNCS*, pages 245–254. Springer, 2006.
12. Z. Chen, B. Fu, J. Xu, B. Yang, Z. Zhao, and B. Zhu. Non-breaking similarity of genomes with gene repetitions. In *Proc. CPM 2007*, volume 4580 of *LNCS*, pages 119–130, 2007.
13. Z. Chen, B. Fu, and B. Zhu. The approximability of the exemplar breakpoint distance problem. In *Proc. AAIM 2006*, volume 4041 of *LNCS*, pages 291–302. Springer, 2006.
14. M. Crochemore, D. Hermelin, G. M. Landau, and S. Vialette. Approximating the 2-interval pattern problem. In *Proc. ESA 2005*, volume 3669 of *LNCS*, pages 426–437. Springer, 2005.
15. H. Fernau. Parameterized algorithms: A graph-theoretic approach. Habilitationsschrift, 2005.
16. M.R. Garey and D.S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. W.H. Freeman, San Franciso, 1979.
17. A. Goldstein, P. Kolman, and Z. Zheng. Minimum common string partition problem: Hardness and approximations. In *Proc. ISAAC 2004*, volume 3341 of *LNCS*, pages 473–484. Springer, 2004.
18. M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
19. P. Kolman and T. Waleń. Reversal distance for strings with duplicates: Linear time approximation using hitting set. In *Proc. WAOA 2006*, volume 4368 of *LNCS*, pages 279–289. Springer, 2006.
20. W. Li, Z. Gu, H. Wang, and A. Nekrutenko. Evolutionary analysis of the human genome. *Nature*, 409:847–849, 2001.
21. M. Marron, K. M. Swenson, and B. M. E. Moret. Genomic distances under deletions and insertions. *Theoretical Computer Science*, 325(3):347–360, 2004.
22. C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

23. D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):909–917, 1999.
24. D. Sankoff and L. Haque. Power boosts for cluster tests. In *Proc. RECOMB-CG 2005*, volume 3678 of *LNCS*, pages 121–130. Springer, 2005.
25. J. Tang and B. M. E. Moret. Phylogenetic reconstruction from gene-rearrangement data with unequal gene content. In *Proc. WADS 2003*, volume 2748 of *LNCS*, pages 37–46. Springer, 2003.