Aligning and Labeling Genomes Under the Duplication-Loss Model

Riccardo Dondi[1] and Nadia El-Mabrouk[2]

[1] DSUS, Università degli Studi di Bergamo, Bergamo - Italy
[2] DIRO, Université de Montréal, Montréal, Québec - Canada
riccardo.dondi@unibg.it, mabrouk@iro.umontreal.ca

Abstract. In this paper we investigate the complexity of two combinatorial problems related to genome alignment, a recent approach to genome comparison based on a duplication-loss model of evolution. The first combinatorial problem, Duplication-Loss Alignment, aims to align two genomes and to explain the unaligned part of the genomes as duplications and losses. The problem has been recently shown to be NP-hard, even when each gene has at most five occurrences in each genome. Here, we improve this result by showing that Duplication-Loss Alignment is APX-hard even if the number of occurrences of a gene inside a genome is bounded by 2. Then we consider a second combinatorial problem, Minimum Relabeling Alignment, and we show that it is equivalent to Minimum Feedback Vertex Set on Direct Graph, hence implying that the problem is APX-hard, is fixed-parameter tractable and approximable within factor $O(\log |\mathcal{X}| \log \log |\mathcal{X}|)$, where $\mathcal{X}$ is the aligned genome considered by Minimum Relabeling Alignment.

1 Introduction

The comparison of complete genomes usually considers two kinds of mutations: (1) macro-evolutionary events such as rearrangements (inversions, transpositions, translocations etc.) and (2) content modifying operations (duplications, losses, horizontal gene transfer etc.) that affect the overall organization of genes. Put differently, usually genomes are represented as strings of symbols over an alphabet Σ of gene families. In the past, genome comparison has been largely based on rearrangement events [3, 9, 12, 16, 4, 15, 17, 18, 7, 8, 11]). Contrariwise, we introduced in [13] an evolutionary model restricted to content-modifying operations (duplications and losses). We showed that this model is effective in studying the evolution of certain gene families, such as Transfer RNAs (tRNAs). From a combinatorial point of view, when rearrangements are ignored gene organization is preserved, hence allowing to reformulate the comparison of two genomes as a Duplication-Loss Alignment problem: find an alignment minimizing the cost of duplications and losses. As in [13], we consider here the cost of an alignment to be the number of underlying segmental duplications (duplication of a string of adjacent genes) and single losses (loss of a single gene).

In this paper, we investigate the complexity of two combinatorial problems related to this approach, Duplication-Loss Alignment and Minimum Relabeling

Alignment. The second problem stems from a direct approach to Duplication-Loss Alignment which is in two steps: (1) Compute a best candidate labeled alignment between the two genomes that may be unfeasible for the Duplication-Loss model and (2) (Minimum Relabeling Alignment problem) Find an evolutionary scenario of minimum duplication-loss cost that is in agreement with the alignment. A similar approach has been proposed in [2], where first it is computed an unlabeled alignment of two genomes, and then the alignment is explained with an evolutionary scenario of minimum duplication-loss.

We show in Section 3 that the Duplication-Loss Alignment is APX-hard, even if the number of occurrences of a gene inside a genome is bounded by 2. Duplication-Loss Alignment is known to be NP-hard in [5] when each gene has at most five occurrences in each genome. Notice that in practice genes have few occurrences inside a genome, so it is interesting to understand how the complexity of the problem is influenced by this parameter. We then show in Section 4 that Minimum Relabeling Alignment is equivalent to Minimum Feedback Vertex Set on Direct Graph, hence showing that the problem is APX-hard and that (1) it is fixed-parameter tractable, when the parameter is the cost of the relabeling, (2) it is approximable within factor $O(\log|\mathcal{X}|\log\log|\mathcal{X}|)$, where $\mathcal{X}$ is the aligned genome considered by Minimum Relabeling Alignment.

2 Preliminaries

Strings: We consider single chromosomal (circular or linear) genomes, hence, given an alphabet Σ, each symbol representing a specific gene family, a *genome* or *string* is a sequence of symbols from Σ, where each symbol may have many occurrences. For example, X in Figure 1 is a genome on the alphabet $\Sigma = \{a, b, c, d, e, f\}$, with four gene copies from the gene family identified by b.

Given a string Z, we denote by $|Z|$ its length, by $Z[i]$, $1 \leq i \leq |Z|$, the i-th symbol of Z, and by $Z[i, j]$, $1 \leq i \leq j \leq |Z|$, the substring of Z that starts at position i and ends at position j. Finally, we say that two substrings $Z[i_1, i_2]$ and $Z[j_1, j_2]$, with $1 \leq i_2 \leq j_2 \leq |Z|$, overlap if $j_1 \leq i_2$.

Graphs: In the rest of the paper we will consider both directed and undirected graphs. An undirected graph is *cubic*, when each of its vertex has degree 3. Consider now a directed graph $G = (V, A)$. Given a vertex $v \in V$, we denote by $IN(v) = \{u \in V : (u, v) \in A\}$. A *feedback vertex set* (FVS) of G is a subset $V' \subseteq V$ such that V' contains at least one vertex from every directed cycle in G.

The Duplication-Loss Model of Evolution: We assume that present-day genomes have evolved from an ancestral string through duplications and losses, where, given a genome X: (i) A *Duplication* of size z is an operation that copies a substring of size z of X somewhere else in the genome. Given two identical non overlapping substrings $X[i, i + z - 1]$ and $X[j, j + z - 1]$ of X, we denote by $D = (X[i, i + z - 1], X[j, j + z - 1])$ a *duplication* from $X[i, i + z - 1]$ to $X[j, j + z - 1]$; $X[i, i + z - 1]$ is the *source*, and $X[j, j + z - 1]$ is the *target* of the duplication D; (ii) A *loss* of size z is an operation $L = (X[i, i + z - 1])$ that removes a substring $X[i, i + z - 1]$ of size z from X.

Given an integer $z \geq 1$, we denote by $c(D(z))$ the cost of a duplication of size z, and by $c(L(z))$ the cost of a loss of size z.

The Duplication-Loss Alignment Problem: We introduced in [13] the concept of "Feasible" Labeled Alignment of two genomes X and Y. Definitions on alignments are given below, and illustrated in Figure 1.

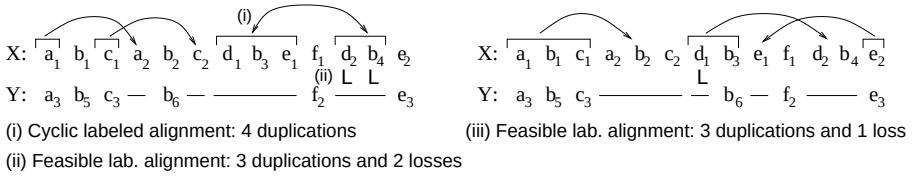

Fig. 1. Alignments for strings X and Y. Costs are $c(D(z)) = 1$ and $c(L(z)) = z$ for any integer z. Losses are denoted by "L" and duplications by arrows from source (indicated by bracket) to target. Two different labeling are given for the left alignment: one (i) with "$d_2\, b_4$" being interpreted as the target of a duplication, and one (ii) with the same substring interpreted as two losses.

In the rest of the paper, we consider two genomes X and Y on an alphabet Σ. Denote with $\Sigma^- = \Sigma \cup \{-\}$ be the alphabet Σ augmented with a symbol '-' (called a *gap*) not in Σ.

Definition 1. *An* Alignment *of X and Y, denoted by $\mathcal{A}(\mathcal{X}, \mathcal{Y})$, consists of a pair $(\mathcal{X}, \mathcal{Y})$ of strings on $\Sigma^- \times \Sigma^-$ obtained by filling X and Y respectively with gaps, such that (1) the* Aligned Genomes *$\mathcal{X}$ and $\mathcal{Y}$ are equal length and (2) for each position i, $1 \leq i \leq |\mathcal{X}|$, it holds that either $\mathcal{X}[i] = \mathcal{Y}[i] \neq -$ (i is called a* Match*), or exactly one of $\mathcal{X}[i]$, $\mathcal{Y}[i]$ is equal to a gap (i is called a* Mismatch*).*

An explanation of an alignment $\mathcal{A}(\mathcal{X}, \mathcal{Y})$ with a duplication-loss history leading to X and Y from a common ancestor,requires a labeling of the mismatched positions of the aligned genomes $\mathcal{X}$ and $\mathcal{Y}$ in terms of *duplications* and *losses*.

Definition 2. *A* Labeling *$\mathcal{L}(\mathcal{X})$ of an aligned genome $\mathcal{X}$ is a set of losses and duplications, such that for each mismatched position j, $1 \leq j \leq |\mathcal{X}|$, $\mathcal{L}(\mathcal{X})$ contains either a loss $L = (\mathcal{X}[j_1, j_2])$ or exactly one duplication $D = (\mathcal{X}[i_1, i_2], \mathcal{X}[j_1, j_2])$ with $1 \leq j_1 \leq j \leq j_2 \leq |\mathcal{X}|$. A* Labeled Alignment *$\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ is a* labeling *of the two aligned genomes $\mathcal{X}$ and $\mathcal{Y}$.*

The cost of a labeling $\mathcal{L}(\mathcal{X})$, denoted by $c(\mathcal{L}(\mathcal{X}))$, is the cost of the underlying operations (losses and duplications). The cost of a labeled alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ is the sum of $c(\mathcal{L}(\mathcal{X}))$ and $c(\mathcal{L}(\mathcal{Y}))$.

A correct interpretation of an alignment in term of duplication-loss history, must prevent from a "cyclic" interpretation of an alignment (see the labeled alignment (i) in Figure 1), where cycles are rigorously defined as follows.

Definition 3. *Consider a set of duplications $\mathcal{D}$. $\mathcal{D}$ induces a* Duplication Cycle *if there is a permutation $D_1 = (\mathcal{X}[i_1, r_1], \mathcal{X}[j_1, s_1])$, $D_2 = (\mathcal{X}[i_2, r_2], \mathcal{X}[j_2, s_2])$, ..., $D_h = (\mathcal{X}[i_h, r_h], \mathcal{X}[j_h, s_h])$ of the duplications in $\mathcal{D}$, such that (1) the substrings $\mathcal{X}[j_p, s_p]$ and $\mathcal{X}[i_{p+1}, r_{p+1}]$ overlap, for each $1 \leq p \leq h-1$, and (2) the substrings $\mathcal{X}[j_h, s_h]$ and $\mathcal{X}[i_1, r_1]$ overlap.*

Now, a labeling $\mathcal{L}(\mathcal{X})$ is Feasible if contains no subset of duplications that induces a duplication cycle. A Feasible Labeled Alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ is a feasible labeling of an alignment of X and Y where $\mathcal{L}(\mathcal{X})$ and $\mathcal{L}(\mathcal{Y})$ are feasible labeling (see Figure 1, (ii) and (iii)).

We are now ready to give the main optimization problem introduced in [13].

Problem 1 ***Duplication-Loss Alignment****[DLA]*
Input*: Two genomes X and Y.*
Output*: A Feasible Labeled Alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ of minimum cost.*

Minimum Feasible Realbeling: A natural approach to DLA proceeds in two steps. First, based on a dynamic programming approach, compute a (possibly cyclic) labeled alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ of minimum cost. Then, the alignment is relabeled in an optimal way, e.g. find feasible labeling $\mathcal{L}'(\mathcal{X})$ and $\mathcal{L}'(\mathcal{Y})$ for $\mathcal{X}$ and $\mathcal{Y}$ respectively, by replacing some of the duplications with losses. Notice that once the genomes are aligned, each feasible relabeling can be computed independently. Given an aligned genome $\mathcal{X}$ and a labeling $\mathcal{L}(\mathcal{X})$, a *feasible relabeling* $\mathcal{L}'(\mathcal{X})$ of $\mathcal{L}(\mathcal{X})$ is a labeling of $\mathcal{X}$ obtained by transforming some duplications of $\mathcal{L}(\mathcal{X})$ into losses, so that $\mathcal{L}'(\mathcal{X})$ is a feasible labeling of $\mathcal{X}$. The *cost* of a feasible relabeling is $c(\mathcal{L}'(\mathcal{X})) - c(\mathcal{L}(\mathcal{X})) = \sum_{D \text{ is relabeled by } \mathcal{L}'(\mathcal{X})}(|D| - 1)$.

Hence, the Minimum Feasible Relabeling problem can be defined as follows:

Problem 2 ***Minimum Feasible Relabeling****[MFR]*
Input*: an aligned genome $\mathcal{X}$ and a labeling $\mathcal{L}(\mathcal{X})$.*
Output*: a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $\mathcal{L}(\mathcal{X})$ having minimum cost.*

3 Complexity of Duplication-Loss Alignment

In this section we investigate the complexity of Duplication-Loss Alignment, and we show that the problem is APX-hard even when each gene appears at most twice in the genome (we denote this restriction as 2-DLA). We prove the APX-hardness of 2-DLA by giving a reduction from Minimum Vertex Cover on Cubic Graphs (MVCC), which is known to be APX-hard [1]. Given an undirected cubic graph $G = (V, E)$, MVCC asks for a subset $V' \subseteq V$ of minimum cardinality, such that for each edge $\{u, v\} \in E$, at least one of u, v is in V'.

Let $G = (V, E)$ be a cubic graph, in the following we define an instance (X, Y) of DLA. Given a vertex v_i and its incident edges $\{v_i, v_j\}$, $\{v_i, v_p\}$, $\{v_i, v_q\}$, with $j < p < q$, we say that $\{v_i, v_j\}$ ($\{v_i, v_p\}$, $\{v_i, v_q\}$ respectively) is the first (second, third respectively) edge incident in v_i.

First, set $t = 9|V|$. We define the alphabet Σ over which X and Y range:

$$\Sigma = \{\alpha_{i,j} : v_i \in V \wedge 1 \leq j \leq 6\} \cup \Gamma \cup \{\beta_{i,j} : v_i \in V \wedge 1 \leq j \leq 4\} \cup \Lambda$$

where

$$\Gamma = \{\gamma_{i,j} : v_i \in V \wedge 1 \leq j \leq t\}, \Lambda = \{\lambda_{i,j,h} : \{v_i, v_j\} \in E \wedge 1 \leq h \leq t\}$$

Now, for each $v_i \in V$, define two substrings $B_X(v_i)$, $B_Y(v_i)$ (substrings of X, Y respectively), as follows:

$$B_X(v_i) = \alpha_{i,1} \dots \alpha_{i,6} \beta_{i,1} \dots \beta_{i,4};\ B_Y(v_i) = \beta_{i,1} \dots \beta_{i,4} \alpha_{i,1} \dots \alpha_{i,6}$$

Now, consider the edge $\{v_i, v_j\} \in E$ and assume that $\{v_i, v_j\}$ is the h-th edges incident in v_i, $1 \leq h \leq 3$, and the k-th edge incident in v_j, $1 \leq k \leq 3$. Define two substrings $B_X(e_{i,j})$, $B_Y(e_{i,j})$ of X, Y respectively, associated with $\{v_i, v_j\}$, as follows:

$$B_X(e_{i,j}) = \alpha_{i,2h-1}\alpha_{i,2h}\alpha_{j,2k-1}\alpha_{j,2k};\ B_Y(e_{i,j}) = \alpha_{j,2k-1}\alpha_{j,2k}\alpha_{i,2h-1}\alpha_{i,2h}$$

Now, we are able to define the two genomes X, Y:

$$X = \gamma_{1,1}\dots\gamma_{1,t}B_X(v_1)\gamma_{2,1}\dots\gamma_{2,t}B_X(v_2)\dots\gamma_{n,1}\dots\gamma_{n,t}B_X(v_n)\cdot \lambda_{1,w,1}\dots\lambda_{1,w,t}B_X(e_{1,w})\dots\lambda_{p,q,1}\dots\lambda_{p,q,t}B_X(e_{p,q})$$

$$Y = \gamma_{1,1}\dots\gamma_{1,t}B_Y(v_1)\gamma_{2,1}\dots\gamma_{2,t}B_Y(v_2)\dots\gamma_{n,1}\dots\gamma_{n,t}B_Y(v_n)\cdot \lambda_{1,w,1}\dots\lambda_{1,w,t}B_Y(e_{1,w})\dots\lambda_{p,q,1}\dots\lambda_{p,q,t}B_Y(e_{p,q})$$

It is easy to see that (X, Y) is an instance of 2-DLA, as each symbol of Σ has at most two occurrences in each of X, Y. Recall that the cost of a duplication of length z is $c(D(z)) = 1$, while the cost of a loss of length z is $c(L(z)) = z$.

In order to prove the main properties of the reduction we have to show some intermediate results. First, in Prop. 1, we show that all the positions containing symbols in $\Gamma \cup \Lambda$ are aligned.

Proposition 1 *Consider an optimal alignment $\mathcal{A}(\mathcal{X}, \mathcal{Y})$ having cost less than $2t$. Then, $\mathcal{A}(\mathcal{X}, \mathcal{Y})$ aligns each position of X and Y containing a symbol in $\Gamma \cup \Lambda$.*

As a consequence of Prop. 1, we can assume that if two positions containing symbols $\Sigma \setminus (\Gamma \cup \Lambda)$ of X, Y are aligned, then either they both belong to substrings $B_X(v_i)$, $B_Y(v_i)$, with $v_i \in V$, or they both belong to substrings $B_X(e_{i,j})$, $B_Y(e_{i,j})$, with $\{v_i, v_j\} \in E$. Now, we present a property of the alignment of the substrings $B_X(v_i)$, $B_Y(v_i)$ and $B_X(e_{i,j})$, $B_Y(e_{i,j})$.

Proposition 2 *Consider the substrings $B_X(v_i)$, $B_Y(v_i)$, $B_X(e_{i,j})$, $B_Y(e_{i,j})$ for some $v_i \in V$ and some $\{v_i.v_j\} \in E$. Then, any alignment of X and Y results in at most 6 matched positions of $B_X(v_i)$, $B_Y(v_i)$ and at most 2 matched positions of $B_X(e_{i,j})$, $B_Y(e_{i,j})$.*

Now, we are ready to prove the main results of the reduction. The idea of the reduction is that for each pair of substrings $B_X(v_i)$, $B_Y(v_i)$ we have two possible cases: either the substrings $\alpha_{i,1}\dots\alpha_{i,6}$ are aligned (corresponding to vertex v_i in the vertex cover of G) or the substrings $\beta_{i,1}\dots\beta_{i,4}$ are aligned (corresponding to vertex v_i not in the vertex cover of G).

Lemma 1. *Let $G = (V, E)$ be a cubic graph and let (X, Y) be the corresponding instance of 2-DLA. Then, given a vertex cover $V' \subseteq V$ of G, we can compute in polynomial time a feasible labeled alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ of cost $8|V'| + 6|V \setminus V'| + 2|E|$.*

Proof. Let V' be a vertex cover of G, we compute a feasible labeled alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ as follows.

For each $v_i \in V'$, align the substrings $\alpha_{i,1}\dots\alpha_{i,6}$ of $B_X(v_i)$, $B_Y(v_i)$ and define a loss for each position containing a symbol $\beta_{i,j}$, $1 \leq j \leq 4$. Hence the cost for aligning $B_X(v_i)$, $B_Y(v_i)$ in this case is 8.

For each $v_i \in V \setminus V'$, align the substrings $\beta_{i,1} \dots \beta_{i,4}$ of $B_X(v_i)$, $B_Y(v_i)$, and for each $\{v_i, v_j\} \in E$, where $\{v_i, v_j\}$ is the h-th edges of v_i, $1 \leq h \leq 3$, define a duplication from the substring $\alpha_{i,2h-1}\alpha_{i,2h}$ of $B_X(e_{i,j})$ ($B_Y(e_{i,j})$ respectively) to the string $\alpha_{i,2h-1}\alpha_{i,2h}$ of $B_X(v_i)$ ($B_Y(v_i)$ respectively). Hence the cost for aligning $B_X(v_i)$, $B_Y(v_i)$ in this case is 6.

Finally, let $\{v_i, v_j\}$ be the h-th edge incident in v_i, and assume that $v_i \in V'$. Align the substrings $\alpha_{j,2k-1}\alpha_{j,2k}$ of $B_X(e_{i,j})$ and $B_Y(e_{i,j})$, and define a duplication from the substring $\alpha_{i,2h-1}\alpha_{i,2h}$ of $B_X(v_i)$ ($B_Y(v_i)$ respectively) to the substring $\alpha_{i,2h-1}\alpha_{i,2h}$ of $B_X(e_{i,j})$ ($B_Y(e_{i,j})$ respectively). The cost for aligning $B_X(e_{i,j})$ and $B_Y(e_{i,j})$ is 2.

By construction, it follows that $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ has a cost of $8|V'| + 6|V \setminus V'| + 2|E|$. □

Lemma 2. *Let $G = (V, E)$ be a cubic graph and let (X, Y) be the corresponding instance of 2-DLA. Then, given an alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ of cost $8p+6(|V|-p)+2|E|$ we can compute in polynomial time a vertex cover of G of size p.*

Proof. (Sketch.) Consider an alignment $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ of cost $8p + 6(|V| - p) + 2|E|$. By construction, by Prop. 1, and by Prop. 2, it can be shown that $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ satisfies the following properties: (1) the alignment of $B_X(e_{i,j})$, $B_Y(e_{i,j})$ has a cost of two, having a duplication either from a substring of $B_X(v_i)$ ($B_Y(v_i)$ respectively) or from a substring of $B_X(v_j)$ ($B_Y(v_j)$ respectively) to a substring of $B_X(e_{i,j})$ ($B_Y(e_{i,j})$ respectively); (2) if there is a duplication from a substring of $B_X(v_i)$ ($B_Y(v_i)$ respectively) to a substring of $B_X(e_{i,j})$ ($B_Y(e_{i,j})$ respectively), then $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ aligns the substrings $\alpha_{i,1} \dots \alpha_{i,6}$ of $B_X(v_i)$, $B_Y(v_i)$, and labels as losses the substrings $\beta_{i,1} \dots \beta_{i,4}$ in $B_X(v_i)$, $B_X(v_j)$; (3) if $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ defines no duplication starting from a substring of $B_X(v_i)$, $B_X(v_j)$, then $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ aligns the substrings $\beta_{i,1} \dots \beta_{i,4}$ in $B_X(v_i)$, $B_Y(v_i)$, and each substring $\alpha_{i,2h-1}\alpha_{i,2h}$ $1 \leq h \leq 3$, of $B_X(v_i)$ (of of $B_Y(v_i)$ respectively), is the target of a duplication from the substring $\alpha_{i,2h-1}\alpha_{i,2h}$ of $B_X(e_{i,j})$ (of $B_Y(e_{i,j})$ respectively), where $\{v_i, v_j\}$ is the h-th edges of v_i.

Hence the set $V' = \{v_i : \alpha_{i,1} \dots \alpha_{i,6}$ is aligned by $\mathcal{A}(\mathcal{L}(\mathcal{X}), \mathcal{L}(\mathcal{Y}))$ in $B_X(v_i)$, $B_y(v_i)\}$ is a vertex cover of G, with $|V'| = p$. □

The APX-hardness of 2-DLA is a direct consequence of Lemmas 1, 2, and of the APX-hardness of MVCC [1].

4 Complexity of Minimum Feasible Relabeling

In what follows we show that MFR is equivalent to the Minimum Directed Feedback Vertex Set (DFVS) problem. Given a directed graph $G = (V, A)$, DFVS asks for a feedback vertex set $V' \subseteq V$ of minimum cardinality.

First, in Section 4.1 we give an L-reduction from DFVS to MFR. As a consequence, we prove that MFR is APX-hard. Then, in Section 4.2, we give a reduction from DFVS to MFR which implies that MFR is fixed-parameter tractable and is approximable within factor $O(\log |\mathcal{X}| \log \log |\mathcal{X}|)$.

4.1 Hardness of MFR

In this section we give an L-reduction from DFVS to MFR. Given a direct graph $G = (V, A)$, with $V = \{v_1, \ldots, v_n\}$, in what follows we define the corresponding genome $\mathcal{X}$ and labeling $\mathcal{L}(\mathcal{X})$. Given a substring s of $\mathcal{X}$, we denote with s^a the fact that s is aligned in $\mathcal{X}$ (hence it does not need any labeling). In the definition of $\mathcal{L}(\mathcal{X})$, first we define the aligned genome $\mathcal{X}$, then we define the labeling of $\mathcal{X}$.

Before giving the details of the construction we give an overview of the construction of $\mathcal{X}$ and $\mathcal{L}(\mathcal{X})$. For each vertex $v_i \in V$ we define a substring $F(v_i)$ obtained by concatenating four substrings $s_{i,IN}$, $s_{i,1}$, $s_{i,2}$, $s_{i,OUT}$ (see Fig. 2). The reduction defines two kind of duplications: (1) duplications between substrings of $F(v_i)$ (one duplication between $s_{i,IN}$, $s_{i,1}$, one duplication between $s_{i,1}$, $s_{i,2}$, one duplication between $s_{i,2}$, $s_{i,OUT}$); (2) duplications between substrings of different $F(v_i)$, $F(v_j)$. The latter kind of duplications encodes arcs of the graph G. Furthermore, notice that $s_{i,IN}$ is used to encode arcs incoming in v_i, while $s_{i,OUT}$ is used to encode arcs outgoing from v_i.

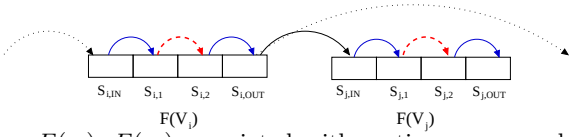

Fig. 2. Substrings $F(v_i)$, $F(v_j)$ associated with vertices v_i, v_j and arc (v_i, v_j); arcs represent duplications. Dashed red arcs are associated with candidate duplications.

Now, we define formally the instance of MFR. Define the alphabet $\Sigma = \{w_{i,j,t} : (v_i, v_j) \in A, 1 \leq t \leq n+2\} \cup \{x_i, y_i : v_i \in V\}$. Given an arc $(v_i, v_j) \in A$, define the string $e_{i,j}$ as follows: $e_{i,j} = w_{i,j,1} w_{i,j,2} \ldots w_{i,j,n+2}$.

Now, we define the strings $s_{i,IN}$, $s_{i,1}$, $s_{i,2}$, $s_{i,OUT}$ (notice that we assume that $IN(v_i) = \{v_{h_1}, , v_{h_z}\}$, and that $h_1 < h_2 \cdots < h_z$):
$s_{i,IN} = e_{i,h_1}\ e_{i,h_2}\ \ldots\ e_{i,h_z}\ x_i^a$; $s_{i,1} = e_{i,h_1}\ e_{i,h_2}\ \ldots\ e_{i,h_z}\ x_i\ y_i^a$;
$s_{i,2} = e^a_{i,h_1}\ e^a_{i,h_2}\ \ldots\ e^a_{i,h_z}\ x_i\ y_i$; $s_{i,OUT} = e_{i,h_1}\ e_{i,h_2}\ \ldots\ e_{i,h_z}\ x_i$.

Define $F(v_i) = s_{i,IN} \cdot s_{i,1} \cdot s_{i,2} \cdot s_{i,OUT}$. The aligned genome $\mathcal{X}$ is defined as follows: $\mathcal{X} = F(v_1) \cdot F(v_2) \cdot \cdots \cdot F(v_n)$.

Now, we define the labeling $\mathcal{L}(\mathcal{X})$ of $\mathcal{X}$. $\mathcal{L}(\mathcal{X})$ consists of two kinds of duplications: duplications between two substrings of the same $F(v_i)$ and duplications between substrings of different sets $F(v_i)$, $F(v_j)$. We start by defining the labeling of the strings in $F(v_i)$ (for the not aligned symbols), which is used to encode the vertex $v_i \in V$:
(1) a duplication from the substring $e_{i,h_1}\ e_{i,h_2}\ \ldots\ e_{i,h_z} x_i^a$ of $s_{i,IN}$ to the substring $e_{i,h_1}\ e_{i,h_2}\ \ldots\ e_{i,h_z} x_i$ of $s_{i,1}$;
(2) a duplication from the substring $x_i\ y_i^a$ of $s_{i,1}$ to the substring $x_i\ y_i$ of $s_{i,2}$;
(3) a duplication from the substring $e^a_{i,h_1}\ e^a_{i,h_2}\ \ldots\ e^a_{i,h_z} x_i$ of $s_{i,2}$ to the substring $e_{i,h_1}\ e_{i,h_2}\ \ldots\ e_{i,h_z} x_i$ of $s_{i,OUT}$.

Now, we define the duplications between substrings of $\mathcal{X}$ that belong to different $F(v_i)$. Those duplications are used to encode the arcs in A. Given an arc $(v_i, v_j) \in A$, define a duplication from the substring $e_{i,j}$ of $s_{i,OUT}$ to the substring $e_{i,j}$ of $s_{j,IN}$.

The duplication from the substring $s_{i,1}$ to the substring $s_{i,2}$, with $1 \leq i \leq n$, both belonging to $F(v_i)$ (notice that the duplicated string is $x_i\ y_i$), is called a *candidate duplication*, and it is denoted by D_i (see Fig. 2). Each other duplication is called a *non candidate duplication*. Informally, the reduction is based on the following properties. Since each non candidate duplication has a cost of at least $n+1$ (it includes the duplication of substring $e_{i,j}$), it follows that: (1) a feasible relabeling $\mathcal{L}'(\mathcal{X})$ is computed by relabeling only candidate duplications (Lemma 3); (2) a vertex v_i in a solution V' of DFVS corresponds to the removal of a candidate duplication D_i. First, we show that we can consider only solutions of MFR that relabel only candidate duplications.

Lemma 3. *Given a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $\mathcal{L}(\mathcal{X})$, we can compute in polynomial time a relabeling $\mathcal{L}''(\mathcal{X})$ of $\mathcal{L}(\mathcal{X})$ such that (1) $\mathcal{L}''(\mathcal{X})$ relabels only candidate duplications, and (2) the cost of $\mathcal{L}''(\mathcal{X})$ is not greater than that of $\mathcal{L}'(\mathcal{X})$.*

Now, we present the two main properties of the reduction.

Lemma 4. *Let $G = (V, E)$ be a directed graph, and let $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ be the corresponding instance of MFR. Then, given a feedback vertex set V' of G, we can compute in polynomial time a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $(\mathcal{X}, \mathcal{L}'(\mathcal{X}))$ of cost $|V'|$.*

Proof. (Sketch.) Let V' be a feedback vertex set of G. Then, we define a solution a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ that relabels the following set of duplications $\{D_i : v_i \in V'\}$ as losses. It is easy to see that $\mathcal{L}'(\mathcal{X})$ is a feasible labeling of $\mathcal{X}$, since V' is a feedback vertex set of G. □

Lemma 5. *Let $G = (V, E)$ be a graph, and let $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ be the corresponding instance of MFR. Then, given a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ of cost c, we can compute in polynomial time a feedback vertex set V' of G, with $|V'| \leq c$.*

The APX-hardness of MFR is a direct consequence of Lemmas 4, 5 and of the APX-hardness of DFVS [14].

4.2 Tractability of MFR

In this section we give give a reduction from MFR to DFVS. The reduction we present is both a parameterized and an approximation preserving reduction, hence it follows that: (1) MFR is fixed-parameter tractable, when parameterized by the cost of the solution; (2) MFR can be approximated within factor $O(\log |\mathcal{X}| \log \log |\mathcal{X}|)$.

Now, let $\mathcal{X}$ be a labeled genome associated with a labeling $\mathcal{L}(\mathcal{X})$. In what follows, we define the directed graph $G = (V, A)$ (input of DFVS) associated with $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$. Consider the set $\mathcal{D}$ of duplications induced by $\mathcal{L}(\mathcal{X})$. First, notice that we assume that each duplication $D \in \mathcal{D}$ has size at least 2, otherwise we can relabel such a duplication with cost 0.

Now, we define $G = (V, A)$ as follows. $V = \bigcup_{D \in \mathcal{D}} V(D)$, where $V(D)$ is a set of vertices associated with duplication $D \in \mathcal{D}$, defined as follows: $V(D) = \{v_{D,i} : 1 \leq i \leq |D| - 1\}$.

Now, we define the set of arcs A:

$$A = \{(v_{D_i,p}, v_{D_j,q}) : D_i = (\mathcal{X}[i_1, i_2], \mathcal{X}[i_3, i_4]) \wedge D_j = (\mathcal{X}[j_1, j_2]\mathcal{X}[j_3, j_4]) \wedge \\ \mathcal{X}[i_3, i_4], \mathcal{X}[j_1, j_2] \text{ overlap}, 1 \leq p \leq |D_i| - 1, 1 \leq j \leq |D_j| - 1\}.$$

Informally, given two duplications D_i, D_j, such that the target of D_i and the source of D_j overlap, we have an arc from each vertex of $V(D_i)$ to each vertex of $V(D_j)$.

Next, we show how to relate a feedback vertex set V' of G and a solution of MFR having size $|V'|$. The idea is that a set $V(D_i)$ of nodes in the feedback vertex set of G corresponds to a duplication D_i relabeled as loss. Notice that a feedback vertex set V' of G is *minimal* if there exists no vertex $v \in V'$ such that $V' \setminus \{v\}$ is a feedback vertex set of G. Next, we prove some properties of a minimal FVS of G.

Lemma 6. *Let V' be a minimal feedback vertex set of the graph $G = (V, E)$ associated with $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$. Then, given a duplication D_i of $\mathcal{D}$, either all the vertices of $V(D_i)$ belong to V' or none of the vertices of $V(D_i)$ belongs to V'.*

Now, we are ready to prove the main properties of the reduction.

Lemma 7. *Let $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ be an instance of MFR and let G be the corresponding instance of DFVS. Then, given a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ of cost c, we can compute in polynomial time a feedback vertex set of G of size c.*

Proof. (Sketch.) Assume that $\mathcal{L}'(\mathcal{X})$ is a feasible relabeling of $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ and let $\mathcal{D}'$ be the set of duplications of $\mathcal{L}(\mathcal{X})$ relabeled as losses by $\mathcal{L}'(\mathcal{X})$. Define the feedback vertex set V' of G as $V' = \bigcup_{D \in \mathcal{D}'} V(D)$. It is easy to see that if V' is not a feedback vertex set, then $\mathcal{L}'(\mathcal{X})$ induces a duplication cycle. Since $|V(D)| = |D| - 1$, it follows that $|V'| = \sum_{D \in \mathcal{D}'} |D| - 1 = c(\mathcal{L}'(\mathcal{X}))$. □

Lemma 8. *Let $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ be an instance of MFR and let G be the corresponding instance of DFVS. Then, given a minimal feedback vertex set V' of G, we can compute in polynomial time a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ of cost $|V'|$.*

Proof. (Sketch.) Assume that V' is a minimal feedback vertex set of G. Since by Lemma 6 either $V(D) \subseteq V'$ or $V(D) \cap V' = \emptyset$, we define a feasible relabeling $\mathcal{L}'(\mathcal{X})$ of $(\mathcal{X}, \mathcal{L}(\mathcal{X}))$ by relabeling as losses the following set of duplications: $\mathcal{D}' = \{D : V(D) \subseteq V'\}$. Since V' is a feedback vertex set of G, it is easy to see that $\mathcal{L}'(\mathcal{X})$ is feasible. Finally, $c(\mathcal{L}'(\mathcal{X})) = \sum_{D \in \mathcal{D}'} |D| - 1 = |V'|$. □

Theorem 3 is a consequence of Lemma 7, Lemma 8, and of the fact that DFVS admits a fixed-parameter algorithm of time complexity $O(4^k k! poly - time(|\mathcal{X}|))$ [6], and it is approximable within factor $O(\log |V| \log \log |V|)$ [19, 10].

Theorem 3 *The MFR problem: (1) admits a fixed-parameter algorithm of time complexity $O(4^k k! poly - time(|\mathcal{X}|))$, where k is the size of the relabeling; (2) admits an approximation algorithm of factor $O(\log |\mathcal{X}| \log \log |\mathcal{X}|)$.*

5 Conclusion

In this paper, we investigated the complexity of two problems, MLA and MFR, related to the alignment of two genomes based on a duplication and loss model of evolution. Interesting future work include the investigation of the approximation and parameterized complexity of DLA.

References

1. Alimonti, P., Kann, V.: Some APX-completeness results for cubic graphs. Theoretical Computer Science 237(1–2), 123–134 (2000)
2. Benzaid, B., Dondi, R., El-Mabrouk, N.: Duplication-loss genome alignment: Complexity and algorithm. In: LATA 2013 (2013)
3. Bergeron, A.: A very elementary presentation of the hannenhalli-pevzner theory. In: CPM 2001. pp. 106–117 (2001)
4. Bourque, G., Pevzner, P.: Genome-scale evolution: Reconstructing gene orders in the ancestral species. Genome Research 12, 26 – 36 (2002)
5. Canzar, S., Andreotti, S.: A branch-and-cut algorithm for the 2-species duplication-loss phylogeny problem. CoRR abs/1208.2698 (2012)
6. Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM 55(5) (2008)
7. El-Mabrouk, N.: Mathematics of Evolution and Phylogeny, chap. Genome rearrangement with gene families, pp. 291- 320. Oxford University Press, Oxford (2005)
8. El-Mabrouk, N., Sankoff, D.: Evolutionary genomics: statistical and computational methods, chap. Analysis of Gene Order Evolution beyond Single-Copy Genes. Methods in Molecular Biology, Springer (Humana), New York (2012)
9. El-Mabrouk, N.: Genome rearrangement by reversals and insertions/deletions of contiguous segments. In: CPM 2000. vol. 1848, pp. 222–234 (2000)
10. Even, G., Naor, J., Schieber, B., Sudan, M.: Approximating minimum feedback sets and multicuts in directed graphs. Algorithmica 20(2), 151–174 (1998)
11. Fertin, G., Labarre, A., Rusu, I., Tannier, E., Vialette, S.: Combinatorics of genome rearrangements. The MIT Press, Cambridge (2009)
12. Hannenhalli, S., Pevzner, P.A.: Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). J. ACM 48, 1–27 (1999)
13. Holloway, P., Swenson, K.M., Ardell, D.H., El-Mabrouk, N.: Evolution of genome organization by duplication and loss: An alignment approach. In: Chor, B. (ed.) RECOMB 2012. pp. 94–112. Springer, Heidelberg (2012)
14. Kann, V.: On the Approximability of NP-complete Optimization Problems. Ph.D. thesis, Royal Institute of Technology of Stockholm (1992)
15. Ma, J., Zhang, L., Suh, B., Raney, B., Burhans, R., Kent, W., Blanchette, M., Haussler, D., Miller, W.: Reconstructing contiguous regions of an ancestral genome. Genome Research 16, 1557 – 1565 (2007)
16. Marron, M., Swenson, K.M., Moret, B.M.E.: Genomic distances under deletions and insertions. In: COCOON 2003. pp. 537–547. Heidelberg (2003)
17. Moret, B., Wang, L., Warnow, T., Wyman, S.: New approaches for reconstructing phylogenies from gene order data. Bioinformatics 17, S165–S173 (2001)
18. Sankoff, D., Blanchette, M.: The median problem for breakpoints in comparative genomics. In: COCOON 1997. pp. 251–264 (1997)
19. Seymour, P.D.: Packing directed circuits fractionally. Combinatorica 15(2), 281–288 (1995)

Appendix

Proofs of Section 3

We show that given a cubic graph $G = (V, E)$, the corresponding pair of genomes (X, Y) is an instance of 2-DLA.

Proposition 4 *Each symbol of Σ has at most two occurrences in each of X, Y.*

Proof. By construction each symbol in $\Gamma \cup \Lambda \cup \{\beta_{i,j} : 1 \leq i \leq |V|, 1 \leq j \leq 4\}$ has exactly one occurrence in each of X and Y. Each symbol $\alpha_{i,j}$, with $\{v_i, vj\} \in E$, has two occurrences in X (Y respectively): one in $B_X(v_i)$ ($B_Y(v_i)$ respectively) and one in $B_X(e_{i,j})$ ($B_Y(e_{i,j})$ respectively). □

Proof of Prop. 1

Proposition 5 *Consider an alignment A of X, Y having cost less than $2t$. Then, A aligns each position of X and Y containing a symbol in $\Gamma \cup \Lambda$.*

Proof. Notice that by construction each symbol in $\Gamma \cup \Lambda$ occurs exactly once in each of X and Y. As a consequence if a position containing a symbol in $\Gamma \cup \Lambda$ is not aligned, it must be a loss. It follows that if all the positions that separates two substrings $B_X(v_i)$, $B_X(v_{i+1})$ and two substrings $B_Y(v_i)$, $B_Y(v_{i+1})$ are not aligned, all such positions are losses, hence the alignment has a cost of at least $2t$. Hence assume that position p in X and position p in Y are aligned, where $X[p] = Y[p] = \gamma_{i,z}$, Let p_l be the position of X and Y such that $X[p_l] = Y[p_l] = \gamma_{i,1}$ and let p_r be the position of X and Y such that $X[p_r] = Y[p_r] = \gamma_{i,t}$. Since p is aligned in X and gy, each position of $X[1, p_l]$ ($X[p_r, |X|]$ respectively) can be aligned only with a position of $gy[1, p_l]$ ($Y[p_r, |Y|]$ respectively). But then, we can modify $\mathcal{A}(X, Y)$ by aligning all the positions containing symbols of $\gamma_{i,y}$, $1 \leq y \leq t$, without modifying any alignment defined by A, hence increasing the number of aligned positions. □

Proof of Prop. 2

Proposition 6 *Consider the substrings $B_X(v_i)$, $B_Y(v_i)$, $B_X(e_{i,j})$, $B_Y(e_{i,j})$ for some $v_i \in V$ and some $\{v_i.v_j\} \in E$. Then, any alignment of X and Y results in at most 6 matched positions of $B_X(v_i)$, $B_Y(v_i)$ and at most 2 matched positions of $B_X(e_{i,j})$, $B_Y(e_{i,j})$.*

Proof. By construction, either the positions of $B_X(v_i)$, $B_Y(v_i)$ containing symbols $\alpha_{i,1} \dots \alpha_{i,6}$ are aligned or positions of $B_X(v_i)$, $B_Y(v_i)$ containing symbols $\beta_{i,1} \dots \beta_{i,4}$ are aligned.

Moreover, by construction, either the positions of $B_X(e_{i,j})$, $B_Y(e_{i,j})$ containing symbols $\alpha_{i,2h-1}\alpha_{i,2h}$ are aligned or positions of $B_X(e_{i,j})$, $B_Y(e_{i,j})$ containing symbols $\alpha_{i,2k-1}\alpha_{i,2k}$ are aligned. □

Proofs of Section 4.1

Proof of Prop. 3

Lemma 9. *Given a feasible relabeling $\mathcal{L}'$ of $\mathcal{X}$, we can compute in polynomial time a relabeling $\mathcal{L}''$ of $\mathcal{X}$ such that (1) each duplication of $\mathcal{L}' \setminus \mathcal{L}''$ is a candidate duplication, and (2) the cost of $\mathcal{L}''$ is not greater than that of $\mathcal{L}'$.*

Proof. Notice that each non candidate duplication duplicates a substring of length at least $n + 2$, since by construction each non candidate duplication duplicates a string containing $e_{i,j}$, and by construction $|e_{i,j}| = n + 2$.

Consider the feasible relabeling $\mathcal{L}'$ and assume that $\mathcal{L}'$ is obtained by removing a non candidate duplication of $\mathcal{L}$ (otherwise $\mathcal{L}''$ is exactly $\mathcal{L}'$). Since by construction each non candidate duplication duplicates at least $n + 2$ positions (hence relabeling this duplication as a loss has a cost of least $n + 1$), we compute $\mathcal{L}''$ starting from $\mathcal{L}'$ as follows: $\mathcal{L}''$ relabels all the candidate duplications of $\mathcal{L}$ as losses, while all the non candidate duplications of $\mathcal{L}$ are not relabeled. Since there are n candidate duplications in $\mathcal{X}$, and each candidate duplication duplicates a string of length 2 (hence relabeling this duplication as a loss has a cost of 1), it follows that the cost of $\mathcal{L}''$ is not greater than the cost of $\mathcal{L}'$.

What is left to show is that $\mathcal{L}''$ is a feasible relabeling (that is $\mathcal{L}''$ induces no cycle). Assume by contradiction that there is a cycle C induced by the labeling $\mathcal{L}''$ of $\mathcal{X}$. By construction $\mathcal{L}$ induces no cycle in the labeling of the substrings of $\mathcal{X}$ in the set $F(v_i)$, $v_i \in V$, and the same property holds for $\mathcal{L}'$. Hence the cycle C must include substrings from at least two different sets $F(v_i)$ and $F(v_j)$, for some $(v_i, v_j) \in E$. It follows, by construction, that C must include a path from $s_{i,IN}$ to some $s_{j,IN}$. But $\mathcal{L}''$ removes all the candidate duplications, hence also the candidate duplication of $F(v_i)$, thus leading to a contradiction. □

Proof of Lemma 6

Lemma 10. *Consider the graph $G = (V, E)$ associated with $(\mathcal{X}, \mathcal{L})$ and let V' be a minimal feedback vertex set of G. Then, given a duplication D of $\mathcal{L}$, either all the vertices of $V(D)$ belong to V' or none of the vertices of $V(D)$ belongs to V'.*

Proof. Assume that a vertex $v_{D,i} \in V$, for some i with $1 \leq i \leq |D| - 1$, is in V', while a vertex $v_{D,j}$, for some $j \neq i$ with $1 \leq j \leq |D|$, is not in V'. Since V' is a feedback vertex set, it follows that $G[V \setminus V']$ does not contain cycles. Denote by $IN(v) = \{u \in V : (u, v) \in A\}$, and by $OUT(v) = \{u \in V : (v, u) \in A\}$. By construction, $IN(v_{D,i}) = IN(v_{D,j})$, and $OUT(v_{D,i}) = OUT(v_{D,j})$, and $\{v_{D,i}, v_{D,j}\} \notin E$, hence $G[(V \setminus V') \cup v_{D,i}]$ will not contain a cycle. Thus $V' \setminus \{v_{D,i}\}$ is a feedback vertex set, and V' could not be a minimal feedback vertex set of G. □

Proof of Theorem 3

Theorem 7 *The MFR problem:*

1. *admits a fixed-parameter algorithm of time complexity* $O(4^k k! poly{-}time(|\mathcal{X}|))$, *where* k *is the size of the relabeling;*
2. *admits an approximation algorithm of factor* $O(\log|\mathcal{X}|\log\log|\mathcal{X}|)$.

Proof. The result follows from Lemma 7 and Lemma 8 and from the following facts.

(1) Notice that the size of the graph G is polynomial in $|\mathcal{X}|$, more precisely $|V| \leq |\mathcal{X}|^2$. Indeed, there exist at most $|\mathcal{X}|$ duplications, as there exists at most $|\mathcal{X}|$ sets $V(D)$ associated with a duplication D of $\mathcal{L}$, and each set $V(D)$ contains at most $\mathcal{X}$ vertices, since each duplication is obviously bounded by the size of $\mathcal{X}$. Since DFVS admits a fixed-parameter algorithm of time complexity $O(4^k k! poly - time(|V|))$, where k is the size of the FVS, it follows that MFR admits a fixed-parameter algorithm of time complexity $O(4^k k! poly - time(|\mathcal{X}|))$, where k is the number of duplications transformed into losses.

(2) Since DFVS admits an approximation algorithm of size $O(\log|V|\log\log|V|)$, and since by the previous argument, $|V| \leq |\mathcal{X}|^2$, MFR admits an approximation algorithm of factor $O(\log|\mathcal{X}|\log\log|\mathcal{X}|)$. □