On the Minimum Error Correction Problem for Haplotype Assembly in Diploid and Polyploid Genomes

Paola Bonizzoni, Riccardo Dondi, Gunnar W. Klau, Yuri Pirola, Nadia Pisanti, Simone Zaccaria

On the Minimum Error Correction Problem for Haplotype Assembly in Diploid and Polyploid Genomes[†]

Authors	E-Mail	Telephone
Paola Bonizzoni[1,⋆]	bonizzoni@disco.unimib.it	(+39)02 6448 7814
Riccardo Dondi[2]	riccardo.dondi@unibg.it	(+39)035 205 2423
Gunnar W. Klau[3,5]	gunnar.klau@cwi.nl	(+31)20 592 4012
Yuri Pirola[1]	pirola@disco.unimib.it	(+39)02 6448 7879
Nadia Pisanti[4,5]	pisanti@di.unipi.it	(+39)050 221 3152
Simone Zaccaria[1]	simone.zaccaria@disco.unimib.it	(+39)02 6448 7917

Addresses

[1] Dip. di Informatica Sistemistica e Comunicazione, Univ. degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milan, Italy.

[2] Dip. di Scienze Umane e Sociali, Univ. degli Studi di Bergamo, Via Donizetti 3, 24129 Bergamo, Italy.

[3] Life Sciences Group, Centrum Wiskunde & Informatica (CWI), Science Park 123, 1098 XG Amsterdam, The Netherlands.

[4] Dip. di Informatica, Univ. degli Studi di Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy.

[5] ERABLE Team, INRIA.

[†] A preliminary version of this paper partially appeared in (Bonizzoni *et al.*, 2015).

Abstract

In diploid genomes, haplotype assembly is the computational problem of reconstructing the two parental copies, called *haplotypes*, of each chromosome starting from sequencing reads, called *fragments*, possibly affected by sequencing errors. *Minimum Error Correction* (MEC) is a prominent computational problem for haplotype assembly and, given a set of fragments, aims at reconstructing the two haplotypes by applying the minimum number of base corrections. MEC is computationally hard to solve, but some approximation-based or fixed-parameter approaches have been proved capable of obtaining accurate results on real data. In this work, we expand the current characterization of the computational complexity of MEC from the approximation and the fixed-parameter tractability point of view. In particular, we show that MEC is not approximable within a constant factor while it is approximable within a logarithmic factor in the size of the input. Furthermore, we answer open questions on the fixed-parameter tractability for parameters of classical or practical interest: the total number of corrections and the fragment length. In addition, we present a direct 2-approximation algorithm for a variant of the problem that has also been applied in the framework of clustering data.

Finally, since *polyploid* genomes, such as those of plants and fishes, are composed of more than two copies of the chromosomes, we introduce a novel formulation of MEC, namely the *k-ploid MEC problem*, that extends the traditional problem to deal with polyploid genomes. We show that the novel formulation is still both computationally hard and hard to approximate. Nonetheless, from the parameterized point of view, we prove that the problem is tractable for parameters of practical interest such as the number of haplotypes and the coverage, or the number of haplotypes and the fragment length.

1 Introduction

The genome of diploid organisms, as humans, is composed of two parental copies, called *haplotypes*, for each chromosome. The most frequent form of genetic variations between the two haplotypes of the same chromosome are the *Single Nucleotide Polymorphisms (SNPs)*. Haplotype analysis is of fundamental importance for a variety of applications including agricultural research, medical diagnostics, and drug design (Pirola *et al.*, 2013; Bonizzoni *et al.*, 2003; Browning and Browning, 2008).

The task of the *haplotyping problem* is the reconstruction of each haplotype. However, large scale direct experimental reconstruction from the collected samples is not yet cost-effective. One of the computational approaches that have been proposed, *haplotype assembly*, considers high-throughput sequencing reads (also called *fragments*) that have to be partitioned in order to reconstruct the haplotypes. Since for most of the SNP positions only two nucleotides are observed, the haplotypes and, hence, the fragments can be represented as binary vectors. The fragments obtained from sequencing may not cover some positions of the haplotypes. These uncovered positions are called *holes*, whereas a sequence of holes within a fragment is called *gap*.

The presence of sequencing and mapping errors makes the haplotype assembly problem a challenging task. In the literature, different combinatorial formulations of the problem have been proposed (Lancia *et al.*, 2001; Lippert *et al.*, 2002; Aguiar and Istrail, 2012; Dondi, 2012). Among them, *Minimum Error Correction* (MEC) (Lippert *et al.*, 2002) has been proved particularly successful in the reconstruction of accurate haplotypes for diploid species (He *et al.*, 2010; Chen *et al.*, 2013; Pirola *et al.*, 2015b). However, MEC is a computationally hard problem. Indeed, MEC is APX-hard even if the fragments have at least one gap (Cilibrasi *et al.*, 2007) and remains NP-hard even if the fragments do not contain gaps

(*Gapless MEC*) (Cilibrasi *et al.*, 2007).

The genome of certain species – especially plants, fishes, and yeasts – is *polyploid*, that is, it is composed of more than two copies for each chromosome, and the analysis of such genomes may improve our knowledge of their specific variants, as well as of the mechanisms of eukaryotic evolution (Berger *et al.*, 2014; Aguiar and Istrail, 2013). Still, the development of haplotype assembly methods for polyploid genomes has received only little attention in the literature (Berger *et al.*, 2014; Aguiar and Istrail, 2013; Das and Vikalo, 2015). In fact, the mathematical foundations and a formulation of the MEC problem for polyploid genomes have not been thoughtfully investigated yet.

The parameterized complexity framework proved to be useful for coping with the computational intractability of MEC on diploid genomes, as it did for several well-known hard combinatorial problems (Downey and Fellows, 2013). In particular, MEC is in FPT when parameterized by the *coverage* (Patterson *et al.*, 2014, 2015), that is, the maximum number of fragments that cover a SNP position. Moreover, MEC is in FPT also when parameterized by the length of the fragments (He *et al.*, 2010), but this is known only under the *all-heterozygous assumption*, which forces to reconstruct complementary haplotypes. In fact, this assumption allows the dynamic programming algorithm of He *et al.* (2010) to focus on the reconstruction of a single haplotype and, hence, to limit the possible combinations for each SNP position. Despite the significant amount of work present in the literature for the diploid case, some important questions related to the fixed-parameter tractability and approximability of MEC are still open. Two significant open problems are whether there exists a constant-factor approximation algorithm for MEC and whether MEC is in FPT when parameterized by parameters of classical or practical interest, such as the total number of corrections or the length of the fragments. Indeed, removing the dependency on the all-heterozygous assumption from the algorithm by He *et al.* (2010) does not appear straightfor-

ward and, hence, fixed-parameter tractability of MEC when parameterized by the fragment length is still an open problem.

The restriction of MEC where the fragments do not contain holes (*Binary MEC*) is particularly interesting from a mathematical point of view, and is the variant of the well-known *Hamming k-Median Clustering Problem* (Cilibrasi *et al.*, 2007; Kleinberg *et al.*, 1998) when $k = 2$. This clustering problem asks for k representative "consensus" (also called "median") strings with the goal of minimizing the Hamming distance between each input string and its closest consensus string. Hamming 2-Median Clustering is well studied from the approximation viewpoint, and at least two Polynomial Time Approximation Schemes (PTAS) have already been proposed (Ostrovsky and Rabani, 2002; Jiao *et al.*, 2004). Instead, the computational complexity of Binary MEC is still unknown.

In this work, we present advances in the characterization of the fixed-parameter tractability and the approximability of MEC problem in the general, gapless, and binary cases. We first show that MEC is not in APX, *i.e.*, it is not approximable within constant factor. In addition, we show that MEC is not in XP when parameterized by the number of non-hole elements on SNP positions and fragments. Since these parameters are upper bounds for the maximum number of corrections on each SNP position and on each fragment, it follows that there is no algorithm for MEC exponential in the maximum number of corrections on each SNP position and on each fragment. These parameters are of particular interest, since recent sequencing tecnologies produce datasets with a low error rate and/or with a uniform distribution of sequencing errors, hence the expected maximum number of corrections to apply on each column/fragment is lower than the coverage/fragment length. However, this result basically rules out the existence of fixed-parameter algorithms on (natural) parameters strictly smaller than coverage and fragment length. Moreover, we show that a reduction previously known (Fouilhoux and Mahjoub, 2012) can be adapted to prove that MEC is ap-

proximable within factor $O(\log nm)$ (where n is the number of fragments and m is the number of SNPs) and that MEC is in FPT when parameterized by the total number of corrections. By inspecting novel combinatorial properties of gapless instances, we also show that Gapless MEC is in FPT when parameterized by the length of the fragments and that Binary MEC can be approximated within factor 2. Although Binary MEC is known to admit a PTAS, the 2-approximation algorithm we give is more practical and intuitive than the previous approximation results. Table 1 summarizes all the known results prior to this work and highlights the novel contributions we present here (some of them were also presented in the preliminary version of this paper (Bonizzoni *et al.*, 2015)), establishing the new state of the art for MEC, Gapless MEC, and Binary MEC problems. Table 1

Furthermore, in this paper we extend the formulation of the MEC problem to the polyploid case by introducing the *k-ploid Minimum Error Correction (k-ploid MEC)* problem and we analyze the aspects regarding its computational complexity, parameterized tractability, and approximability. Notice that SNP positions usually assume at most two values also in the polyploid case and, as a consequence, the haplotypes and the fragments can be still represented as binary vectors. In particular, since fixed-parameter tractable algorithms for parameters of practical interest revealed to be successful for dealing with diploid genomes, we show that k-ploid MEC is in FPT when parameterized by the coverage and the number of haplotypes and when parameterized by the fragment length and the number of haplotypes. The latter result clearly applies also to the diploid case, but the algorithm that constructively proves this result has a worse time complexity than the one we specifically propose for the diploid case (which is instead based on a theoretical result that does not extend to the polyploid case). Table 2 reports current knowledge of computational complexity, approximability, and fixed-parameter tractability for the newly introduced k-ploid MEC and its variants, k-ploid Gapless MEC and k-ploid Binary MEC. Table 2

Related work

The MEC problem was introduced in (Lippert *et al.*, 2002), where it was shown to be NP-hard on arbitrary instances. Cilibrasi *et al.* (2007) refined the computational complexity analysis by showing that MEC is NP-hard even on instances where fragments do not have gaps (Gapless MEC) and that it is APX-hard on instances where fragments have at most one gap (1-gap MEC). These restrictions are motivated by the characteristics of the prevailing sequencing technologies of that time. Moreover, they also showed that MEC on instances without holes (Binary MEC) is a special form of the Hamming 2-Median Clustering problem and, hence, it admits a Polynomial Time Approximation Scheme (PTAS) in a randomized (Ostrovsky and Rabani, 2002) and deterministic (Jiao *et al.*, 2004) form. Interestingly, the existence of a PTAS for Gapless MEC (or its APX-hardness) and the NP-hardness of Binary MEC are still open questions (albeit Binary MEC with an arbitrary number of haplotypes is NP-hard (Cilibrasi *et al.*, 2007)).

Several heuristic approaches have been proposed to cope with the computational intractability of MEC (see, for example, those surveyed by Geraci (2010) and Duitama *et al.* (2012)). Many of these are based on graph-theoretical formulations of the problem. For example, HapCUT (Bansal and Bafna, 2008) proceeds by iteratively computing max-cuts in a graph where each vertex represents a fragment and each edge denotes the presence of conflicts between fragments, while HapCompass (Aguiar and Istrail, 2012) models the problem as the *Minimum Weighted Edge Removal (MWER)* problem on a particular graph-based representation of the fragments, called *compass graph*, and heuristically solves it with a strategy based on cycle basis local optimization.

Also exact approaches have been successfully proposed. One of the first exact approaches was proposed by Wang *et al.* (2005) who presented an exact algorithm based on the branch-and-bound method. However, this approach was not always suitable for instances of realis-

tic size, and a genetic algorithm was applied as heuristic for those cases. Since fragments produced by Next-Generation Sequencing technologies usually span only a few SNP positions, He *et al.* (2010) proposed a dynamic programming algorithm whose time complexity is exponential in the fragment length. This algorithm, from a theoretical point of view, establishes that MEC is in FPT when parameterized by the fragment length (but only under the all-heterozygous assumption). However, it is also important to deal with long fragments, as they usually improve the accuracy of the solution. As a consequence, the authors also presented a reduction of MEC to MaxSAT in order to use well-known and effective MaxSAT solvers for dealing with realistic instances composed of long fragments. For the same reason, Chen *et al.* (2013) proposed an approach based on Integer Linear Programming (ILP) coupled with a procedure for decomposing the input into small independent blocks in order to improve performances. Notably, this approach does not necessarily rely on the all-heterozygous assumption. Despite the decomposition procedure allows to greatly simplify the input matrix for unweighted instances, the ILP formulation requires a large (quadratic) number of variables and, hence, the approach failed in solving certain blocks of the input, called *hard blocks*, resorting to an heuristic for solving them.

Patterson *et al.* (2014, 2015) proposed an FPT algorithm for MEC when parameterized by the coverage. Using coverage as parameter is motivated by the fact that it is not expected to grow as fast as fragment length in a realistic dataset with the advent of future-generation sequencing technologies. However, limiting the coverage poses a practical limit on datasets that can be managed, even if a recent parallel version (Aldinucci *et al.*, 2014) allowed to obtain a constant factor improvement on coverages that can be handled. To better model the characteristics of data produced by future-generation sequencing technologies, Pirola *et al.* (2015b) recently proposed a novel variant of the MEC problem, namely the *k-constrained MEC* problem, where the maximum number of corrections for each column is bounded by a

given constant k, and showed that this variant is in FPT when parameterized by k and coverage by introducing a dynamic-programming algorithm called HapCol (Pirola *et al.*, 2015a). This result does not conflict with the parameterized intractability we present in Section 3 when the parameter is the maximum number of corrections on each column, since HapCol is exponential in both k and coverage. Furthermore, despite an algorithm exponential only in the coverage already existed, HapCol represents a significant practical advancement since the combined use of the two parameters allows to greatly reduce time and space requirements on real data by better modelling the characteristics of such data.

To the best of our knowledge, a theoretical analysis of the computational complexity of the MEC problem on polyploid genomes has never been performed before this work. However, some approaches for dealing with haplotype assembly in polyploid genomes have already been introduced. Aguiar and Istrail (2013) extended HapCompass in order to reconstruct multiple copies of the chromosomes, while Das and Vikalo (2015) formulated the problem as a semi-definite program and devised a fast approximate algorithm for finding a low-rank solution for them. Recently, Berger *et al.* (2014) proposed a maximum-likelihood estimation framework for polyploid haplotype assembly which is able to manage high ploidity while maintaining acceptable performance.

A computational problem related to polyploid haplotype assembly is the sequence multiassembly problem, which is the problem of reconstructing a set of k sequences from their aligned fragments where k is unknown. However, the two problems differ in some key points. Indeed, in the multiassembly problem, the fragments are often assumed to be error-free (or previously error-corrected) and the aim is to minimize the cardinality k of such a set. This computational problem models, for example, the tasks of estimating viral quasispecies (Eriksson *et al.*, 2008) or transcriptome assembly (Trapnell *et al.*, 2010; Song and Florea, 2013; Bao *et al.*, 2013) and is often based on the combinatorial problem of Minimum Path

Cover of a directed acyclic graph representing the overlaps between the fragments. Interestingly, the problem can be solved in polynomial time if the input fragments are gapless (Fulkerson, 1956) or if there are only contiguous subpath constraints (Bao *et al.*, 2013; Rizzi *et al.*, 2014), but becomes NP-hard if the fragments have at least one gap (Rizzi *et al.*, 2014; Beerenwinkel *et al.*, 2014, 2015).

2 Preliminary Definitions

In this section, we introduce some basic notions and the formal definition of the MEC problem. In the rest of the work, we indicate, as usual, the value of a vector s at position t as $s[t]$.

A *fragment matrix* is a matrix $\mathcal{M}$ composed of n rows and m columns such that each entry contains a value in $\{0, 1, -\}$. Each row of $\mathcal{M}$ represents a *fragment* and, formally, is a vector belonging to $\{0, 1, -\}^m$. Symmetrically, each column of $\mathcal{M}$ corresponds to an SNP position and is a vector belonging to $\{0, 1, -\}^n$. We denote by f_i the i-th row of $\mathcal{M}$ and by p_j the j-th column of $\mathcal{M}$. As a consequence, the entry of $\mathcal{M}$ at the i-th row and j-th column is denoted by $f_i[j]$ or $p_j[i]$. The *length* ℓ_i of a fragment f_i is defined as the number of elements in f_i between the rightmost and the leftmost non-hole elements (included) and we denote by ℓ the maximum length over all the fragments in $\mathcal{M}$. Moreover, we say that a column p_j *covers* a row f_i if $p_j[i] \in \{0, 1\}$ or there exist l, r with $l < j < r$ such that $p_l[i], p_r[i] \in \{0, 1\}$ (*i.e.*, $p_j[i]$ is a hole belonging to a gap) and we define the *active fragments* of p_j as the set $\mathcal{A}_{\mathcal{F}}(p_j)$ of all the fragments covered by p_j. We denote by $\mathcal{A}_{\mathcal{F}}(p_{j_1}, p_{j_2})$ the intersection $\mathcal{A}_{\mathcal{F}}(p_{j_1}) \cap \mathcal{A}_{\mathcal{F}}(p_{j_2})$ for two columns p_{j_1} and p_{j_2}. Therefore, the *coverage* cov_j of a column p_j is equal to $|\mathcal{A}_{\mathcal{F}}(p_j)|$ and we define as *cov* the maximum coverage over all the columns of $\mathcal{M}$. A column p_j is *heterozygous* if it contains both 0's and 1's, otherwise is *homozygous*. A *hole* is an entry $f_i[j]$ of $\mathcal{M}$ equal to the symbol $-$. A *gap* in a fragment f_i is a maximal subvector of holes in f_i surrounded by non-hole entries (that is, there exist two positions j_1 and j_2 with $j_1 + 1 < j_2$ such that $f_i[j_1], f_i[j_2] \neq -$ and $f_i[t] = -$ for all t with $j_1 < t < j_2$). A fragment matrix is *gapless* if no fragment contains a gap.

Two rows f_{i_1} and f_{i_2} are in *conflict* when there exists a position j, with $1 \leq j \leq m$, such that $f_{i_1}[j] \neq f_{i_2}[j]$, and $f_{i_1}[j], f_{i_2}[j] \neq -$. Otherwise, we say that f_{i_1} and f_{i_2} are in

agreement. A collection $\mathcal{F}$ of fragments is in *agreement* if any pair of fragments f_1, f_2 in $\mathcal{F}$ are in agreement. A fragment matrix $\mathcal{M}$ is *conflict free* if there exists a bipartition $(\mathcal{F}_1, \mathcal{F}_2)$ of its fragments such that both $\mathcal{F}_1$ and $\mathcal{F}_2$ are in agreement.

When a fragment matrix $\mathcal{M}$ is conflict free, all the fragments in each part of the bipartition can be merged in order to reconstruct a haplotype, intended as a fragment without holes. Unfortunately, a fragment matrix $\mathcal{M}$ is not always conflict free. The Minimum Error Correction problem deals precisely with this issue by asking for a minimum set of *corrections* that make a fragment matrix conflict free, where a correction of a given fragment f_i at position j, with $f_i[j] \neq -$, is the flip of the value $f_i[j]$, replacing a 0 with a 1, or a 1 with a 0.

Problem 1. Minimum Error Correction (MEC) problem
Input: a fragment matrix $\mathcal{M}$ of n rows and m columns.
Output: a conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ with the minimum number of corrections.

Gapless MEC is the restriction of MEC where the input fragment matrix $\mathcal{M}$ is gapless, while *Binary MEC* is the restriction of (Gapless) MEC where the matrix $\mathcal{M}$ does not contain holes (that is, when $\mathcal{M}$ is a binary matrix).

Given a conflict free fragment matrix $\mathcal{M}$, any heterozygous column p_j encodes a bipartition of the fragments covered by p_j indicating which one belongs to one haplotype and which one belongs to other. Instead, any homozygous column p_j gives no information on how the covered fragments have to be partitioned, and it is "in accordance" with any other bipartition or heterozygous column. More formally, we say that two columns p_{j_1}, p_{j_2} of a fragment matrix are in *accordance* if (1) at least one of p_{j_1}, p_{j_2} is homozygous, or (2) p_{j_1}, p_{j_2} are both heterozygous and are identical or complementary on the fragments covered by both.

As stated in the following lemma, pairwise column accordance on gapless matrices is a

necessary and sufficient condition for being conflict free.

Lemma 1. *Let $\mathcal{M}$ be a* gapless *fragment matrix. Then, $\mathcal{M}$ is conflict free if and only if each pair of columns is in accordance.*

Proof. By definition, if $\mathcal{M}$ is conflict free, each pair of columns is in accordance. For this reason, we just prove by induction on the number m of columns in $\mathcal{M}$ that if each pair of columns is in accordance, then $\mathcal{M}$ is conflict free.

If $m = 1$, the lemma obviously holds.

Assume by induction that the lemma holds for the first $m - 1$ columns in $\mathcal{M}$, we need to prove that the lemma still holds for all the m columns. The submatrix on the first $m - 1$ columns is conflict free by induction and, for this reason, a bipartition (P_1, P_2) of the corresponding fragments exists. We assume that p_m is heterozygous, since the lemma clearly holds when p_m is homozygous. Moreover, we define p_h the rightmost heterozygous column on the first $m - 1$ columns and we ignore the homozygous columns between p_h and p_m because they cannot induce conflicts and are in accordance with any other heterozygous column. By assumption, p_h and p_m are in accordance. Hence, p_h and p_m define the same bipartition on the fragments in $\mathcal{A}_{\mathcal{F}}(p_h, p_m)$. Since $\mathcal{M}$ is gapless, there is no column p_y in $\{p_1, \ldots, p_{h-1}\}$ such that $\mathcal{A}_{\mathcal{F}}(p_y, p_m) \setminus \mathcal{A}_{\mathcal{F}}(p_h) \neq \emptyset$, hence $\mathcal{A}_{\mathcal{F}}(p_m) \setminus \mathcal{A}_{\mathcal{F}}(p_h) \not\subseteq \mathcal{A}_{\mathcal{F}}(p_y)$ for $1 \leq y \leq h - 1$. It follows that there exists a bipartition $(P_1 \cup P'_1, P_2 \cup P'_2)$ for every fragment active on all the m columns, where (P'_1, P'_2) is the bipartition induced by p_m on the fragments in $\mathcal{A}_{\mathcal{F}}(p_m) \setminus \mathcal{A}_{\mathcal{F}}(p_{m-1})$. As a consequence the submatrix on the first m columns is conflict free. □

Since such a property is independent by the order of the columns, this result also applies to fragment matrices that can be transformed to gapless matrices by rearranging their columns (*gapless-reducible* fragment matrices). Testing if a fragment matrix $\mathcal{M}$ is gapless-reducible can be performed in polynomial time by testing if the binary matrix $\mathcal{B}(\mathcal{M})$ obtained

from $\mathcal{M}$ by substituting each non-hole element with a one and each hole with a zero has the *consecutive ones property* (C1P) (Meidanis *et al.*, 1998). Therefore, we immediately obtain the following result.

Corollary 2. *Let $\mathcal{M}$ be gapless-reducible fragment matrix. Then, $\mathcal{M}$ is conflict free if and only if each pair of columns is in accordance.*

Gapless-reducibility, beside being of theoretical interest, can also be relevant in practice, as it is potentially able to transform an "almost gapless" fragment matrix (*i.e.*, a fragment matrix with only a few short gaps due, for example, to indel sequencing errors) to a gapless matrix and, hence, to apply algorithms designed for gapless instances which, in general, could be more efficient than those designed for general instances.

Notice that the accordance relation among heterozygous columns is transitive since it basically requires that pairs of columns are equal or complementary. Therefore, since homozygous columns cannot induce conflicts, we have the following result.

Corollary 3. *Let $\mathcal{M}$ be a* gapless *fragment matrix. Then, $\mathcal{M}$ is conflict free if and only if each pair of consecutive columns in the matrix obtained from $\mathcal{M}$ by removing its homozygous columns is in accordance.*

The property defined in Lemma 1 is particularly important when designing exact algorithms for Gapless MEC, as it allows to test only for pairwise column accordance in order to ensure that the matrix is conflict free. In fact, the fixed-parameter algorithm for Gapless MEC that we present in Sect. 4 is based on this property. Furthermore, notice that if we relax the requirement that $\mathcal{M}$ is gapless (or gapless-reducible), then the property does not hold. Consider, for example, the fragment matrix $\mathcal{M}$ composed of three fragments $f_1 = 01-$, $f_2 = -01$, and $f_3 = 1 - 0$. The three columns are pairwise in accordance, but the matrix is not conflict free (and, in fact, f_3 contains a gap).

Given two columns p_{j_1}, p_{j_2} of a fragment matrix $\mathcal{M}$, we define their (generalized) Hamming distance $d_H(p_{j_1}, p_{j_2})$ as $|\{i \mid \{p_{j_1}[i], p_{j_2}[i]\} = \{0, 1\}\}|$ while their *correction distance* $d(p_{j_1}, p_{j_2})$ as the minimum between $d_H(p_{j_1}, p_{j_2})$ and $d_H(\overline{p_{j_1}}, p_{j_2})$ (where $\overline{p}$ is the complement of p on non-hole entries). Notice that the correction distance is non-negative and symmetric, but does not satisfy the triangle inequality, hence, despite the name, is not a metric. We also define the *homozygous distance* $H(p_j)$ as the minimum between the number of 0's and 1's contained in p_j. Intuitively, the correction distance is the cost of making a column equal or complementary to another column, while the homozygous distance is the cost of making a column homozygous.

A solution of MEC over a fragment matrix $\mathcal{M}$ is a bipartition of its fragments, that can be encoded as a binary vector O. It is easy to see that the cost of that solution is:

$$cost_{\mathcal{M}}(O) = \sum_{j=1}^{m} \min(d(O, p_j), H(p_j)). \tag{1}$$

3 Approximation and parameterized complexity of MEC

In this section, we show that MEC is not in APX, that is MEC cannot be approximated within constant factor. We achieve this result by introducing an L-reduction (Ausiello *et al.*, 1999) from the Edge Bipartization problem to MEC.

The Edge Bipartization problem is defined as follows.

Problem 2. Edge Bipartization (EB) problem (Garey and Johnson, 1979)
Input: an undirected graph $G = (V, E)$.
Output: $E' \subseteq E$ of minimum size such that $G' = (V, E \setminus E')$ is bipartite.

Now, we present the details of the reduction. Given an undirected graph $G = (V, E)$, we build the associated fragment matrix $\mathcal{M}(G)$ (with $|V|$ rows and $|E|$ columns) by setting, at each column p_j associated with edge $e_j = \{u, v\} \in E$, $f_u[j] = 0$, $f_v[j] = 1$, and $f_z[j] = -$ for $z \neq u, v$. Notice that, by construction, there exists a conflict in $\mathcal{M}(G)$ between fragments f_u and f_v if and only if $\{u, v\} \in E$.

Lemma 4. *Let $G = (V, E)$ be an undirected graph and $\mathcal{M}(G)$ be the associated fragment matrix. Given a solution E' of EB over G, we can compute in polynomial time a solution of MEC over $\mathcal{M}(G)$ with $|E'|$ corrections. Symmetrically, given a solution of MEC over $\mathcal{M}(G)$ with h corrections, we can compute in polynomial time a solution E' of EB over G of size at most h.*

Proof. ($\Rightarrow$) Let E' be a set of edges such that $(V_1 \uplus V_2, E \setminus E')$ is bipartite, where V_1 and V_2 are the parts of the bipartition. Build a matrix $\mathcal{M}'(G)$ from $\mathcal{M}(G)$ by flipping, for each $e_j = \{u, v\} \in E'$, the entry $f_u[j]$. Clearly, $\mathcal{M}'(G)$ is obtained from $\mathcal{M}(G)$ with $|E'|$ corrections and it does not contain conflicts induced by edges in E'. Let $(\mathcal{F}_1, \mathcal{F}_2)$ be the bipartition of fragments of $\mathcal{M}'(G)$ such that $\mathcal{F}_i := \{f_u \mid v_u \in V_i\}$ (for $i \in \{1, 2\}$). Each $\mathcal{F}_i$ is in agreement

because it does not contain a pair of fragments associated with the endpoints of an edge of $E \setminus E'$. Hence, $\mathcal{M}'(G)$ is conflict free.

($\Leftarrow$) Let $\mathcal{M}'(G)$ be a conflict free matrix obtained from $\mathcal{M}(G)$ with h corrections and let C' be the subset of columns of $\mathcal{M}'(G)$ that contain exactly a correction. In fact, notice that a single correction is sufficient to transform a column into a homozygous column in accordance with any other column. Consider the set $E' := \{e_j \in E \mid p_j \in C'\}$. Clearly, $|E'| \leq h$. Since $\mathcal{M}'(G)$ is conflict free, there exists a bipartition $(\mathcal{F}_1, \mathcal{F}_2)$ of the fragments such that both $\mathcal{F}_1, \mathcal{F}_2$ are in agreement. Build sets V_1, V_2 such that $V_i := \{v_u \mid f_u \in \mathcal{F}_i\}$ (with $i \in \{1, 2\}$). We claim that $(V_1 \uplus V_2, E \setminus E')$ is bipartite. Suppose to the contrary that there exists an edge $e_j = \{u, v\} \in E \setminus E'$ such that $u, v \in V_i$, $i \in \{1, 2\}$. Since $f_u[j] = f_v[j]$ in $\mathcal{M}'(G)$, this implies that exactly one of $f_u[j]$ and $f_v[j]$ has been corrected (since $f_u[j] \neq f_v[j]$ in $\mathcal{M}(G)$). As a consequence, we have that $e_j \in E'$, contradicting the assumption. □

Khot (2002) proved that, under the Unique Games Conjecture, EB is not in APX. Since Lemma 4 proves that MEC is L-reducible to EB, we have the following result.

Theorem 5. *Under the Unique Games Conjecture (Khot, 2002), MEC is not in* APX.

Edge Bipartization is NP-hard even if the graph is cubic, that is if each vertex has degree three (Yannakakis, 1978). If the graph is cubic, then the fragment matrix built by our reduction contains two non-hole elements on each column and three non-hole elements on each fragment. As a consequence, any optimal solution clearly places at most two corrections on each column (actually, at most a correction is placed on each column of any optimal solution, since it is enough for transforming the column into a homozygous column) and at most three corrections on each fragment. Hence, from this observation and from the NP-hardness of EB on cubic graphs, we obtain the following result.

Theorem 6. *MEC is not in* XP[1] *when parameterized by the number of non-hole elements on columns (SNP positions) and on rows (fragments).*

As a consequence of Theorem 6, there is no algorithm for MEC of time complexity $O(n^{f(c_p,c_f)})$ where c_p, c_f are the maximum number of non-hole entries in each column and row, respectively. Furthermore, since the number of non-hole elements on each column and each row are upper bounds for the maximum number of corrections on each column and each fragment, it follows that MEC is not in XP (hence, also not in FPT) when parameterized by these parameters.

The inapproximability result given in Theorem 5 nicely complements an approximation (and fixed-parameter tractable) result that can be inferred by a reduction presented in (Fouilhoux and Mahjoub, 2012), where MEC is reduced to the Maximum Bipartite Induced Subgraph problem (MBIS). Given a vertex-weighted graph G, MBIS asks for a maximum weight subset of vertices of G that induces a bipartite graph. The reduction defines a graph, called *fragment graph*, whose set of nodes is the union of two sets: a set of nodes, called *fragment nodes*, one for each fragment, and a set of nodes, called *entry nodes*, one for each entry of the matrix. In order to avoid the removal of fragments nodes, they are assigned a sufficiently large weight.

The reduction can be easily reworked in order to prove approximation and fixed-parameter tractability results for MEC. More precisely, MEC is now reduced to the *Graph Bipartization* (GB) problem, a problem related to MBIS. Given an unweighted graph G, GB asks for the minimum number of vertex removals so that the resulting graph is bipartite. The reduction given in (Fouilhoux and Mahjoub, 2012) can be modified by defining a new version of the fragment graph (see Fig. 1), where each (weighted) fragment node is substituted with a suf-

[1] We recall that XP is the class of parameterized problems that admit an algorithm of time complexity $n^{f(k)}$ for some computable function f and parameter k.

ficiently large set of fragment nodes. From the construction of the fragment graph, it follows that a fragment matrix $\mathcal{M}$ is conflict free if and only if the corresponding fragment graph is bipartite and that a solution of MEC with k corrections corresponds to a solution of GB that removes k vertices.

Figure 1

Since GB can be approximated within factor $O(\log |V|)$ (Garg *et al.*, 1996) and is in FPT when parameterized by the number of removed vertices (Reed *et al.*, 2004; Guo *et al.*, 2006), we have that:

Theorem 7. *(1) MEC can be approximated in polynomial time within factor* $O(\log nm)$ *where n is the number of rows and m is the number of columns.*
(2) MEC is in FPT *when parameterized by the total number of corrections.*

4 Gapless MEC is in FPT when parameterized by the fragment length

In this section, we introduce a fixed-parameter tractable algorithm for Gapless MEC when parameterized by the maximum length ℓ of the fragments. The algorithm is based on a dynamic programming approach and aims at finding a specific tripartition for the columns of a gapless fragment matrix $\mathcal{M}$. In this section, we assume w.l.o.g. that $\mathcal{M}$ is a gapless fragment matrix and the fragments of $\mathcal{M}$ are sorted by starting position.

Lemma 1 provides a sufficient and necessary condition for the reconstruction of a solution for Gapless MEC, that is a conflict free fragment matrix. For this reason, the gapless condition is required by this algorithm. In fact, if the fragment matrix contains gaps, the accordance of the columns is not sufficient to ensure that there are no conflicts. Therefore, we firstly show a result that directly derives from Lemma 1. In particular, the following proposition stresses the relationship between a bipartition of the fragments and a tripartition of the columns in a gapless fragment matrix $\mathcal{M}$ that is conflict free.

Lemma 8. *Given a gapless fragment matrix $\mathcal{M}$, the following assertions are equivalent:*

1. *$\mathcal{M}$ is conflict free.*
2. *There exists a bipartition $(\mathcal{F}_1, \mathcal{F}_2)$ of the fragments, where both $\mathcal{F}_1$ and $\mathcal{F}_2$ are in agreement.*
3. *There exists a tripartition $T = (L, H, R)$ of the columns such that each column in H is homozygous, each column in $L \cup R$ is heterozygous, $d_H(p_{j_1}, p_{j_2}) = 0$ for all the columns $p_{j_1}, p_{j_2} \in L$ ($p_{j_1}, p_{j_2} \in R$, resp.) and $d_H(\overline{p_{j_1}}, p_{j_2}) = 0$ for each column $p_{j_1} \in L$ and each column $p_{j_2} \in R$.*

Proof. The equivalence between (1) and (2) holds by definition. Therefore, we only show that (1) and (3) are equivalent.

($\Rightarrow$) If $\mathcal{M}$ is conflict free, each pair of columns p_{j_1}, p_{j_2} is in accordance by Lemma 1. By definition, either at least one column is homozygous or $d(p_{j_1}, p_{j_2}) = 0$. It directly follows that a tripartition $T = (L, H, R)$ can be built such that each column in H is homozygous, each column in $L \cup R$ is heterozygous, $d_H(p_{j_1}, p_{j_2}) = 0$ for all the columns $p_{j_1}, p_{j_2} \in L$ ($p_{j_1}, p_{j_2} \in R$, resp.) and $d_H(\overline{p_{j_1}}, p_{j_2}) = 0$ for each column $p_{j_1} \in L$ and each column $p_{j_2} \in R$.

($\Leftarrow$) Let p_{j_1}, p_{j_2} be two columns. If at least one column belongs to H, then p_{j_1} and p_{j_2} are in accordance by definition. Otherwise, when p_{j_1} and p_{j_2} are both heterozygous, then $d(p_{j_1}, p_{j_2}) = 0$. Indeed, if they belong to the same part (L or R), then $d_H(p_{j_1}, p_{j_2}) = 0$, whereas if they belong to different parts then $d_H(\overline{p_{j_1}}, p_{j_2}) = 0$. Hence, p_{j_1} are p_{j_2} in accordance. □

Based on Lemma 8, we introduce an algorithm for Gapless MEC that builds a tripartition of the columns of $\mathcal{M}$ in order to find a conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ with the minimum number of corrections. Notice that in the rest of this section we implicitly refer only to tripartitions built as reported in the third assertion of Lemma 8.

The algorithm iteratively proceeds row-wise and, at each step, computes a tripartition for the columns considered so far. In particular, the key observation that allows to bound the exponential complexity of the algorithm to the parameter ℓ is that we can build any tripartition for all the columns in $\mathcal{M}$ by adding only a subset of columns, called *active columns*, for each row. This subset contains the columns covering the current fragment and the columns covering both previous and successive fragments. Indeed, we need to remember the tripartition established by previous fragments for columns that are covered by successive fragments. More formally, we define the set *active columns* for a fragment f_i as:

$$\mathcal{A}_{\mathcal{C}}(f_i) = \{p_j \mid (p_j[i] \neq -) \vee (\exists x, y \text{ with } x < i < y \mid p_j[x], p_j[y] \neq -)\}$$

Fig. 2 represents the active columns $\mathcal{A}_C(f_i)$ of a fragment f_i. The cardinality of $\mathcal{A}_C(f_i)$ is bounded by ℓ. In fact, considering a row f_i, notice that $\ell_i \leq \ell$ and no column p_k, to the left of f_i, is in $\mathcal{A}_C(f_i)$. Recall that fragments are sorted by starting position and assume that r is the number of columns p_j to the right of f_i, such that there are f_b, f_q with $b < i < q$ and $p_j[b], p_j[q] \neq -$. Since the r columns must be contained in $\mathcal{A}_C(f_b)$ for a fragment f_b with a starting position preceding the one of f_i, it holds that $\ell_i + r \leq \ell_b \leq \ell$. It clearly follows that $|\mathcal{A}_C(f_i)| = \ell_i + r \leq \ell$.

Figure 2

Considering two rows f_{i_1} and f_{i_2}, with $i_1 < i_2$, a tripartition for all the columns in $\mathcal{A}_C(f_{i_1}) \cup \mathcal{A}_C(f_{i_2})$ can be computed by combining a tripartition T_1 for $\mathcal{A}_C(f_{i_1})$ and a tripartition T_2 for $\mathcal{A}_C(f_{i_2})$, only if T_1 and T_2 are "in accordance", that is, they are partitioning the shared columns in the same way. For this reason, we say that a tripartition $T_2 = (L_2, H_2, R_2)$ for $\mathcal{A}_C(f_{i_2})$ *extends* another tripartition $T_1 = (L_1, H_1, R_1)$ for $\mathcal{A}_C(f_{i_1})$ if and only if $L_1 \cap \mathcal{A}_C(f_{i_2}) \subseteq L_2$, $H_1 \cap \mathcal{A}_C(f_{i_2}) \subseteq H_2$, and $R_1 \cap \mathcal{A}_C(f_{i_2}) \subseteq R_2$.

At each step i, the algorithm computes a tripartition T for $\mathcal{A}_C(f_i)$ extending a tripartition T' for $\mathcal{A}_C(f_{i-1})$. Since $\mathcal{A}_C(f_{i-1})$ also contains all the columns p_j with $p_j[i-1] = -$ such that there exists $y < i - 1$ with $p_j[y] \neq -$ and $p_j[i] \neq -$, it follows that T even extends any tripartition computed at the previous steps extended by T'. As a consequence, we prove the following implication.

Lemma 9. *If there exists a conflict free matrix $\mathcal{M}''$ obtained from $\mathcal{M}$ on the first $i-1$ rows that induces a tripartition T' for the columns in $\mathcal{A}_C(f_{i-1})$, and if T is a tripartition for the columns in $\mathcal{A}_C(f_i)$ extending T', then there exists a conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ on the first i rows that induces the tripartition T for the columns in $\mathcal{A}_C(f_i)$.*

Proof. By definition, $p_j[i] \neq -$ and $p_j[y] = -$ for each column $p_j \in \mathcal{A}_C(f_i) \setminus \mathcal{A}_C(f_{i-1})$ and for each $y < i$. By assumption T extends T', hence build $\mathcal{M}'$ such that the columns covered by the first $i-1$ rows are tripartitioned as in $\mathcal{M}''$ and the remaining columns only covered

by f_i are tripartitioned according to T. By construction, $\mathcal{M}'$ induces the tripartition T for $\mathcal{A}_\mathcal{C}(f_i)$. Since $\mathcal{M}''$ is conflict free, it follows that $\mathcal{M}'$ is conflict free by Lemma 8. □

At each step i and for each tripartition $T = (L, H, R)$ for $\mathcal{A}_\mathcal{C}(f_i)$, the algorithm chooses the tripartition T' extended by T for $\mathcal{A}_\mathcal{C}(f_{i-1})$ that induces the minimum cost (*recursive step*) and computes the minimum number of corrections to add on the current fragment f_i in order to tripartition all the columns in $\mathcal{A}_\mathcal{C}(f_i)$ according to T (*local contribution*). In particular, the algorithm considers the minimum number of corrections on f_i such that $p_j[i] = 1$ or $p_j[i] = 0$ for all p_j in L and, on the contrary, $p_j[i] = 0$ or $p_j[i] = 1$ for all p_j in R. At the same time, the minimum number of corrections on the fragment f_i is computed for each column p_j in H such that p_j on the first i rows can be optimally transformed into a homozygous column. Therefore, we define $D[i, T]$ as the minimum number of corrections to obtain a conflict free matrix $\mathcal{M}'$ from $\mathcal{M}$ on the first i rows that induces a tripartition T for $\mathcal{A}_\mathcal{C}(f_i)$. The algorithm proceeds row-wise computing the value $D[i, T]$ for each fragment f_i and for each tripartition T for $\mathcal{A}_\mathcal{C}(f_i)$ by the following recursive equation:

$$D[i, T] = \Delta(i, T) + \min_{T' \text{ extended by } T} D[i-1, T'] \tag{2}$$

where T' is a tripartition for $\mathcal{A}_\mathcal{C}(f_{i-1})$. In the recursion, we consider only the tripartitions T' extended by T, since the shared columns have to be partitioned in the same way. In conclusion, the local contribution is defined as:

$$\Delta(i, T) = O(i, H) + \min \begin{cases} E^0(i, L) + E^1(i, R) \\ E^1(i, L) + E^0(i, R) \end{cases} \qquad \textit{where } T = (L, H, R) \tag{3}$$

such that $E^x(i, F)$ is the cost of correcting the columns in F for fragment f_i to value x, that is $E^x(i, F) = |\{j \mid j \in F \wedge p_j[i] \notin \{x, -\}\}|$, and $O(i, H)$ is the minimum number of corrections to apply on fragment f_i such that the columns in H, considered on the first i rows, can be turned into homozygous columns with minimum cost. Denote by $\#^x_{i,j}$ the number of values

equal to x in $\{p_j[1], \ldots, p_j[i]\}$. The minimum between $\#^0_{i,j}$ and $\#^1_{i,j}$ states the minimum number of corrections necessary to turn a column p_j on the first i rows into a homozygous column. Since $O(i, H)$ refers only to the corrections on fragment f_i, we can compute $O(i, H)$ as:

$$O(i, H) = \sum_{j \in H} \begin{cases} 1 & p_j[i] = 0 \text{ and } \#^0_{i,j} \leq \#^1_{i,j} \\ 1 & p_j[i] = 1 \text{ and } \#^1_{i,j} \leq \#^0_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Given a set of columns F, it is easy to see that $\sum_{i \in \{1,\ldots,n\}} O(i, F) = \sum_{p_j \in F} H(p_j)$.

The base case of the recurrence is $D[1, T] = \Delta(1, T)$ for each tripartition T for $\mathcal{A}_\mathcal{C}(f_1)$. The algorithm returns the optimum corresponding to $\min_T D[n, T]$ where T is a tripartition for $\mathcal{A}_\mathcal{C}(f_n)$. Furthermore, an optimal tripartition for all the columns can be computed by backtracking.

The algorithm computes all the values $D[i, T]$ for each tripartition T of the columns in $\mathcal{A}_\mathcal{C}(f_i)$ and for each i in $\{1, \ldots, n\}$. It follows that there are $O(3^\ell \cdot n)$ entries and, therefore, the space complexity is equal to $O(3^\ell \cdot n)$. Given a tripartition T, we need $O(3^\ell)$ time to enumerate all the tripartitions T' extended by T because we have to tripartition all the columns in $|\mathcal{A}_\mathcal{C}(f_{i-1}) \setminus \mathcal{A}_\mathcal{C}(f_i)|$ with $\mathcal{A}_\mathcal{C}(f_{i-1}) \leq \ell$ and, consequently, $|\mathcal{A}_\mathcal{C}(f_{i-1}) \setminus \mathcal{A}_\mathcal{C}(f_i)| \leq \ell$. Since $\Delta(i, T)$ can be computed in $O(\ell)$ time, each entry $D[i, T]$ can be computed in $O(3^\ell \cdot \ell)$. It follows that the total running time of the algorithm is $O(3^{2\ell} \cdot \ell \cdot n)$. Notice that storing partial information during the computation (using an approach similar to the one presented in (Patterson *et al.*, 2014)) we can decrease the complexity to $O(3^\ell \cdot \ell \cdot n)$.

We now show the correctness of the algorithm.

Lemma 10. *Consider a gapless fragment matrix $\mathcal{M}$.*

1. *If $D[i, T] = h$, then there exists a conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ on the first i rows with h corrections that induces a tripartition T for the columns in $\mathcal{A}_\mathcal{C}(f_i)$.*

2. *If $\mathcal{M}'$ is a conflict free matrix obtained from $\mathcal{M}$ on the first i rows with h corrections that induces a tripartition T for the columns in $\mathcal{A}_{\mathcal{C}}(f_i)$, $D[i,T] \leq h$.*

Proof. We prove the lemma by induction on i. Both the statements obviously hold for $i = 1$. Assume that lemma holds for $i-1$, we show that both the statements hold for i.

(1) By Eq. (2), there exists a tripartition T' for $\mathcal{A}_{\mathcal{C}}(f_{i-1})$ such that T extends T' and $D[i,T] = h = \Delta(i,T) + D[i-1,T']$. Assuming $D[i-1,T'] = h'$, by induction there exists a conflict free matrix $\mathcal{M}''$ obtained from $\mathcal{M}$ on the first $i-1$ rows with h' corrections that induces a tripartition T' for $\mathcal{A}_{\mathcal{C}}(f_{i-1})$. By Proposition 9, there exists a conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ on the first i rows that induces a tripartition T for $\mathcal{A}_{\mathcal{C}}(f_i)$. Since T extends T', by construction we can add $\Delta(i,T)$ corrections on fragment f_i in order to build $\mathcal{M}'$ starting from $\mathcal{M}''$. It follows that $\mathcal{M}'$ is obtained from $\mathcal{M}$ with $\Delta(i,T) + h' = h$ corrections.

(2) Assume that $\mathcal{M}''$ is the submatrix of $\mathcal{M}'$ obtained from $\mathcal{M}$ on the first $i-1$ rows with h' corrections that induces a tripartition T' for $\mathcal{A}_{\mathcal{C}}(f_{i-1})$. Clearly, T' is extended by T due to the fact that $\mathcal{M}''$ is equal to $\mathcal{M}'$ on the first $i-1$ rows. Since $\mathcal{M}'$ contains $\Delta(i,T)$ corrections on the row f_i by construction, it follows that $h = \Delta(i,T) + h'$. Moreover, we know that $D[i-1,T'] \leq h'$ by induction and by Eq. (2) that $D[i,T] = \Delta(i,T) + \min_{T'' \text{ extended by } T} D[i-1,T'']$. Hence, since $\min_{T'' \text{ extended by } T} D[i-1,T''] \leq D[i-1,T']$, we conclude that $D[i,T] \leq \Delta(i,T) + h'$ and, consequently, $D[i,T] \leq h$. □

From the correctness of the algorithm, it directly follows that:

Theorem 11. *Gapless MEC (without the all-heterozygous assumption) is in* FPT *when parameterized by the length of the fragments and it can be solved in $O(3^{\ell} \cdot \ell \cdot n)$ time.*

5 A 2-approximation algorithm for Binary MEC

In this section we present a 2-approximation algorithm for Binary MEC, that is the restriction of MEC where the fragment matrix has values in $\{0, 1\}$ only, and hence does not contain holes. The approximation algorithm is based on the observation that heterozygous columns in binary matrices naturally encode bipartitions of the fragments and that, by Lemma 1, if the columns of a gapless fragment matrix are pairwise in accordance then the matrix is conflict free. In particular, Algorithm 1 builds a feasible solution $\texttt{SOL}[t]$ for each t in $\{1, \ldots, m\}$ assuming that p_t is the closest column to an (unknown) optimal bipartition O of the fragments. Each solution $\texttt{SOL}[t]$ corrects columns $p_{j'}$ with cost $H(p_{j'}) \leq d(p_t, p_{j'})$ into homozygous columns (equal to $\underline{1}$ or $\underline{0}$ depending on the best choice), whereas it corrects the remaining columns $p_{j''}$ with cost $d(p_t, p_{j''}) < H(p_{j''})$ into heterozygous columns equal (or complementary, depending on the best choice) to p_t. It is easy to see that $\texttt{SOL}[t]$ for each t in $\{1, \ldots, m\}$ is a feasible solution (by Lemma 1) and that its cost is exactly $cost_{\mathcal{M}}(p_t)$, as reported in Eq. 1.

Algorithm 1 A 2-approximation algorithm for Binary MEC

Require: A $n \times m$ binary matrix $\mathcal{M}$

```
for t = 1 to m do                ▷ Assume that p_t is the column "closest" to O
    for j = 1 to m do
        if H(p_j) ≤ d(p_t, p_j) then
            Set p_j homozygous in SOL[t]
        else
            Set p_j equal/complementary to p_t in SOL[t]
return arg min_{SOL[t]} cost_M(p_t)
```

Algorithm 1 is a 2-approximation algorithm for Binary MEC.

Lemma 12. *Given a fragment matrix $\mathcal{M}$ without holes, if OPT is the optimum for Binary*

MEC on input $\mathcal{M}$, then Algorithm 1 returns in $O(m^2n)$ time a feasible solution with cost OPT' such that $OPT' \leq 2 \cdot OPT$.

Proof. Assume that p_O is the column of $\mathcal{M}$ closest to an optimal bipartition O, that is $d(O, p_O) \leq d(O, p_j)$ for each j in $\{1, \ldots, m\}$ and assume that $d_H(O, p_O) \leq d_H(\overline{O}, p_O)$ (if $d_H(\overline{O}, p_O) < d_H(O, p_O)$ we can substitute O with $\overline{O}$ since they encode the same bipartition). Clearly, one such a column exists and $d_H(O, p_O) \leq d(O, p_j)$ for each j in $\{1, \ldots, m\}$. We show that, under this assumption, $d(p_O, p_j) \leq 2d(O, p_j)$. By the triangle inequality, $d_H(p_O, p_j) \leq d_H(p_O, O) + d_H(O, p_j)$. Hence, since $d_H(p_O, O) \leq d(O, p_j) \leq d_H(O, p_j)$, we have $d_H(p_O, p_j) \leq 2d_H(O, p_j)$. Similarly, we can prove that $d_H(p_O, \overline{p_j}) \leq 2d_H(O, \overline{p_j})$. As a consequence we have that $d(p_O, p_j) \leq 2d_H(O, p_j)$ and that $d(p_O, p_j) \leq 2d_H(O, \overline{p_j})$, which then imply $d(p_O, p_j) \leq 2d(O, p_j)$. Clearly, since $d(p_O, p_j) \leq 2d(O, p_j)$, we also have that $\min(d(p_O, p_j), H(p_j)) \leq 2\min(d(O, p_j), H(p_j))$.

Since Algorithm 1 iteratively assumes that each column p_j is the closest column to the unknown optimal bipartition O, we have that the cost of the returned solution is $OPT' \leq cost_{\mathcal{M}}(p_O) \leq 2\sum_{j=1}^{m} \min(d(O, p_j), H(p_j)) = 2OPT$. Since each iteration t of the algorithm computes $\texttt{SOL}[t]$ in $O(mn)$ time, the total running time is clearly equal to $O(m^2n)$. □

Algorithm 1 runs in $O(m^2n)$ time and, due to its simplicity, it is a more direct and practical approach than the PTAS algorithms known in the literature (Ostrovsky and Rabani, 2002; Jiao *et al.*, 2004).

6 Polyploid MEC

In this section, we introduce a formulation of the MEC problem applied to polyploid genomes. In particular, we assume that the number k of chromosome copies is known *a priori*, and, for this reason, we consider the k-ploid variant of the problem, that is, the *k-ploid Minimum Error Correction (k-ploid MEC)* problem.

The main concepts and definitions for this problem are the same of those introduced in Section 2 for the (diploid) MEC problem. The most important difference lies in the definition of the output. In fact, the goal of the novel formulation is to reconstruct k haplotypes, where k is the (given) number of chromosome copies that compose the polyploid genome of the species under study. As a consequence, we need to generalize the concept of conflict free fragment matrix. In particular, let $\mathcal{M}$ be a fragment matrix, we say that $\mathcal{M}$ is *k-conflict free* if and only if there exists a k-partition $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_k)$ of its fragments such that each part $\mathcal{F}_i$ is in agreement. At last, we formally define the k-ploid MEC problem:

Problem 3. k-ploid Minimum Error Correction (k-ploid MEC) problem
Input: an integer k and a fragment matrix $\mathcal{M}$ of n rows and m columns.
Output: a k-conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ with the minimum number of corrections.

As for the MEC problem, Gapless k-ploid MEC is the restriction of k-ploid MEC where the input fragment matrix $\mathcal{M}$ is gapless, while Binary k-ploid MEC is the restriction of (Gapless) k-ploid MEC where the matrix $\mathcal{M}$ does not contain holes (that is, when $\mathcal{M}$ is a binary matrix). Furthermore, notice that all the fragments in each part of the k-partition of a k-conflict free matrix can be merged in order to reconstruct a haplotype, in the same way as it can be done for the bipartition of a conflict free matrix in the diploid case.

Clearly, k-ploid MEC is a generalization of (diploid) MEC. As a consequence, k-ploid

MEC inherits some hardness results from MEC. Since we know that MEC is APX-hard (Cilibrasi *et al.*, 2007), that MEC is not in APX under the Unique Games Conjecture (Section 3), that gapless MEC is NP-hard (Cilibrasi *et al.*, 2007), and that MEC is not in XP when parameterized by the number of non-hole elements on rows and columns (Section 3), the following theorem clearly holds.

Theorem 13. *k-ploid MEC is* APX*-hard, k-ploid MEC is not in* APX *under the Unique Games Conjecture, gapless k-ploid MEC is* NP*-hard, and k-ploid MEC is not in* XP *when parameterized by the number k of haplotypes and the number of non-hole elements on rows and columns.*

In addition, Cilibrasi *et al.* (2007) showed that if the number of haplotypes to be reconstructed is specified as part of the input and if the input mfragment matrix does not have holes (such as in the case of k-ploid Binary MEC), the k-ploid Binary MEC problem becomes NP-hard. However, the authors were not able to say whether there exists a constant-factor approximation algorithm for the problem.

In the following sections, we propose two algorithms that allow to prove that k-ploid MEC is in FPT when parameterized by coverage and number of haplotypes (Section 6.1) and when parameterized by fragment length and number of haplotypes (Section 6.2). The choice of parameters is reasonable, since, depending on the characteristics of sequencing technologies, coverage or fragment length are usually limited by small constants. Moreover, species that naturally have more than 8 haplotypes are quite rare among higher-order organisms.

6.1 k-ploid MEC is in FPT when parameterized by coverage and number of haplotypes

In this section, we introduce a fixed-parameter tractable algorithm for k-ploid MEC when parameterized by the coverage *cov* and the number k of haplotypes. The algorithm is based on a dynamic programming approach and aims at finding a k-partition of minimum cost for all the fragments. This algorithm extends on the k-ploid case the general idea used in (Patterson *et al.*, 2014) for solving the diploid MEC problem.

The algorithm iteratively proceeds column-wise and, at each step, computes a k-partition for the fragments covered by the columns considered so far. The key idea for the fixed-parameter tractability of the algorithm is to show that an optimal k-partition for all the fragments can be built through this iterative procedure by adding, at each step, only a subset of fragments whose cardinality is limited by the coverage.

Let $\mathcal{M}$ be a fragment matrix and p_j be one of its columns. Then, we denote with $\mathcal{F}^j = (\mathcal{F}^j_1, \ldots, \mathcal{F}^j_k)$ a k-partition for the fragments in $\mathcal{A}_\mathcal{F}(p_j)$. Let p_{j_1}, p_{j_2} be two columns of $\mathcal{M}$, then we say that $\mathcal{F}^{j_2}$ *extends* $\mathcal{F}^{j_1}$ if and only if, for each $e \in \{1, \ldots, k\}$, $\mathcal{F}^{j_1}_e \cap \mathcal{A}_\mathcal{F}(p_{j_2}) \subseteq \mathcal{F}^{j_2}_e$. If a k-partition $\mathcal{F}^{j_1}$ is extended by a k-partition $\mathcal{F}^{j_2}$, it follows that we can easily build a k-partition $\mathcal{F}^{j_1,j_2}$ for the fragments in $\mathcal{A}_\mathcal{F}(p_{j_1}, p_{j_2})$ such that $\mathcal{F}^{j_1,j_2}$ extends both $\mathcal{F}^{j_1}$ and $\mathcal{F}^{j_2}$. Therefore, it is easy to see that any k-partition for all the fragments covered by the first j columns can be built starting from a k-partition for all the fragments covered by the first $j-1$ columns that induces a k-partition $\mathcal{F}^{j-1}$ for $\mathcal{A}_\mathcal{F}(j-1)$ and a k-partition $\mathcal{F}^j$ for the fragments in $\mathcal{A}_\mathcal{F}(p_j)$ where F^j extends $\mathcal{F}^{j-1}$.

We define $D[j, \mathcal{F}^j]$ as the minimum number of corrections to obtain a k-conflict free matrix $\mathcal{M}'$ from $\mathcal{M}$ on the first j columns such that $\mathcal{M}'$ induces a k-partition $\mathcal{F}^j$ for the fragments in $\mathcal{A}_\mathcal{F}(p_j)$. The value $D[j, \mathcal{F}^j]$ for each column p_j and for each k-partition $\mathcal{F}^j$ for

$\mathcal{A}_{\mathcal{F}}(p_j)$ can be computed by the following recursive equation:

$$D[j, \mathcal{F}^j] = \Delta(j, \mathcal{F}^j) + \min_{\mathcal{F}^{j-1} \text{ extended by } \mathcal{F}^j} D[j-1, \mathcal{F}^{j-1}] \tag{5}$$

where $\mathcal{F}^{j-1}$ is a k-partition for $\mathcal{A}_{\mathcal{F}}(p_{j-1})$ and $\Delta(j, \mathcal{F}^j)$ is the "local contribution" of the k-partition $\mathcal{F}_j$ on column p_j. Informally, $\Delta(j, \mathcal{F}^j)$ is the minimum number of corrections needed for making the fragments in $\mathcal{A}_{\mathcal{F}}(p_j)$ k-conflict free on column p_j according to the k-partition $\mathcal{F}^j$. Such a cost can be easily computed as $\Delta(j, \mathcal{F}^j) = \sum_{e=1}^{k} \min(\#^0(\mathcal{F}_e^j), \#^1(\mathcal{F}_e^j))$ where $\#^u(\mathcal{F}_e^j) = |\{i \mid p_j[i] = u\}|$, *i.e.* the number of fragments in $\mathcal{F}_e^j$ with value u in column p_j.

The base case of the recurrence is $D[1, \mathcal{F}^1] = \Delta(1, \mathcal{F}^1)$ for each k-partition $\mathcal{F}^1$ for the fragments in $\mathcal{A}_{\mathcal{F}}(p_1)$. The optimum is $\min_{\mathcal{F}^m} D[m, \mathcal{F}^m]$ and a corresponding optimal k-partition for all the fragments of the input fragment matrix can be computed by backtracking.

The algorithm computes the entries $D[j, \mathcal{F}^j]$ for each k-partition $\mathcal{F}^j$ of the fragments in $\mathcal{A}_{\mathcal{F}}(p_j)$ and for each j in $\{1, \ldots, m\}$. Since the number of k-partitions for cov elements is k^{cov}, it follows that there are $O(k^{cov}\, m)$ entries. Given a k-partition $\mathcal{F}^j$, we need $O^*(k^{cov})$ time[2] to enumerate all the k-partitions $\mathcal{F}^{j-1}$ extended by $\mathcal{F}^j$ because we have to partition all the fragments in the set $\mathcal{A}_{\mathcal{F}}(p_{j-1}) \setminus \mathcal{A}_{\mathcal{F}}(p_j)$ (whose cardinality is clearly, at most, cov). Since $\Delta(j, \mathcal{F}^j)$ can be computed in polynomial time, each entry $D[j, \mathcal{F}^j]$ can be computed in $O^*(k^{cov})$. It follows that the total running time of the algorithm is $O^*(k^{2cov})$. Notice that, by storing partial information during the computation (as suggested in Section 4), we can decrease the complexity to $O^*(k^{cov})$. We omitted a detailed analysis of the polynomial factors in the time complexity since it would require to explicitly describe how fragments and partitions are represented and manipulated but it would not be useful for our purpose of characterizing the parameterized complexity of the problem.

In conclusion, we prove the correctness of the algorithm.

[2]We recall that $O^*(k^{cov})$ denotes the class $O(k^{cov}\text{poly}(nm))$, where nm is the size of the input.

Lemma 14. *Consider a fragment matrix $\mathcal{M}$.*

1. *If $D[j, \mathcal{F}^j] = g$, then there exists a k-conflict free matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ on the first j columns with g corrections that induces a partition $\mathcal{F}^j$ for the fragments in $\mathcal{A}_{\mathcal{F}}(p_j)$.*

2. *If $\mathcal{M}'$ is a k-conflict free matrix obtained from $\mathcal{M}$ on the first j columns with g corrections that induces a partition $\mathcal{F}^j$ for the fragments in $\mathcal{A}_{\mathcal{F}}(p_j)$, then $D[j, \mathcal{F}^j] \leq g$.*

Proof. We prove the lemma by induction on j. Both statements obviously hold for $j = 1$. Assume that lemma holds for $j - 1$, we show that both statements also hold for j.

(1) By Eq. (5), there exists a k-partition $\mathcal{F}^{j-1}$ for fragments in $\mathcal{A}_{\mathcal{F}}(p_{j-1})$ such that $\mathcal{F}^{j-1}$ is extended by $\mathcal{F}^j$ and $D[j, \mathcal{F}^j] = \Delta(j, \mathcal{F}^j) + D[j-1, \mathcal{F}^{j-1}]$. By induction there exists a k-conflict free matrix $\mathcal{M}''$ obtained from $\mathcal{M}$ on the first $j-1$ columns with at least $D[j-1, \mathcal{F}^{j-1}]$ corrections that induces a partition $\mathcal{F}^{j-1}$ for the fragments in $\mathcal{A}_{\mathcal{F}}(p_{j-1})$. Since $\mathcal{F}^{j-1}$ is extended by $\mathcal{F}^j$, there exists a k-partition $\mathcal{F}^{j-1,j}$ that induces $\mathcal{F}^{j-1}$ when restricted to $\mathcal{A}_{\mathcal{F}}(p_{j-1})$ and that induces $\mathcal{F}^j$ when restricted to $\mathcal{A}_{\mathcal{F}}(p_j)$. As a consequence, we can build a fragment matrix $\mathcal{M}'$ which is equal to $\mathcal{M}''$ on the first $j-1$ columns and whose j-th column is the correction of p_j such that each part of $\mathcal{F}^j$ is in accordance. Such a correction, as explained before, needs to flip at least $\Delta(j, \mathcal{F}^j)$ elements of p_j. Since $\mathcal{M}''$ is k-conflict free and since no fragment in $\mathcal{A}_{\mathcal{F}}(p_j) \setminus \mathcal{A}_{\mathcal{F}}(p_{j-1})$ is covered by one of the first $j-1$ columns, $\mathcal{M}'$ is k-conflict free. Moreover, the total number of corrections needed to obtain $\mathcal{M}'$ from the first j columns of $\mathcal{M}$ is clearly equal to $\Delta(j, \mathcal{F}^j) + D[j-1, \mathcal{F}^{j-1}] = D[j, \mathcal{F}^j]$ by construction.

(2) Assume that $\mathcal{M}''$ is the submatrix of $\mathcal{M}'$ obtained from $\mathcal{M}$ on the first $j-1$ columns with g' corrections that induces a partition $\mathcal{F}^{j-1}$ for the fragments in $\mathcal{A}_{\mathcal{F}}(p_{j-1})$. Obviously, since $\mathcal{M}''$ is equal to the first $j-1$ columns of $\mathcal{M}'$, $\mathcal{M}''$ is k-conflict free and $\mathcal{F}^j$ extends $\mathcal{F}^{j-1}$. Since $\Delta(j, \mathcal{F}^j)$ is the minimum number of corrections needed to transform column p_j into column p'_j of $\mathcal{M}'$ such that each part of $\mathcal{F}^j$ is in agreement, we have that $\Delta(j, \mathcal{F}^j) + g' \leq g$.

By induction we know that $D[j-1, \mathcal{F}^{j-1}] \leq g'$, hence we have that $D[j, \mathcal{F}^{j}] \leq \Delta(j, \mathcal{F}^{j}) + D[j-1, \mathcal{F}^{j-1}] \leq g$. □

From the correctness of the algorithm, it directly follows that:

Theorem 15. *k-ploid MEC is in* FPT *when parameterized by coverage and number of haplotypes.*

6.2 *k*-ploid MEC is in FPT when parameterized by fragment length and number of haplotypes

In this section, we introduce a fixed-parameter tractable algorithm for k-ploid MEC when parameterized by the fragment length ℓ and the number of haplotypes k. Recall that, for many applications, both parameters are bounded by small constants. For example, the widespread Illumina sequencing technologies produce reads spanning only one or a few SNP positions and most species do not have more than 4–8 haplotypes.

This algorithm is based on a different approach than that presented in Section 4 for diploid MEC because that algorithm heavily relies on the characterization of conflict free fragment matrices given by Lemma 8 that cannot be easily extended to the k-ploid case. Indeed, one of the key ingredients for proving the existence of the tripartition T in Lemma 8 (assertion 3) is that each heterozygous column implicitly encodes a partial solution (that is, a bipartition of the fragments covered by that column). As a consequence, the existence of a solution for the fragments covered by any pair of heterozygous columns (i.e., their accordance) can be easily checked by testing the equality or the complementarity between the two columns on their shared active fragments. This idea cannot be directly applied to the k-ploid case, since a single heterozygous column is clearly not sufficient to encode a k-partition (with k non-empty parts). Groups of h columns (for some fixed h) could be a natural encoding of

k-partitions, but, in this case, we cannot check their accordance only by testing the equality or complementarity between these groups (as we do in the diploid case). Hence, a characterization of k-conflict free fragment matrices based on the existence of a partition for the columns (analogous to the tripartition T of the diploid case) seems not straightforward. Notice that the algorithm we introduce in this section for k-ploid MEC can be clearly applied also to diploid MEC, but the one designed in Section 4 has a better time complexity.

The novel algorithm uses an approach similar to that of (He *et al.*, 2010). In particular, the algorithm, rather than k-partitioning the fragments, aims at the direct reconstruction of k haplotypes such that each fragment aligns to one of them with the minimum total amount of mismatches. Clearly, this is equivalent to computing a k-conflict free fragment matrix $\mathcal{M}'$ obtained from $\mathcal{M}$ with the minimum amount of corrections. Indeed, given k haplotypes, a k-conflict free matrix $\mathcal{M}'$ can be computed in polynomial time by correcting each fragment in a such of way that it perfectly aligns to its closest haplotype (in terms of Hamming distance on non-hole elements). Intuitively, the algorithm, for each column p_j, computes the minimum number of mismatches needed for aligning each fragment ending at column p_j or before to a set of k haplotypes. Such a minimum number of mismatches is computed by partitioning the fragments into two parts: (*i*) those ending exactly at column p_j and (*ii*) those ending strictly before (that is, on the left of) column p_j. The key observation for reducing the time complexity is that the set of fragments ending at column p_j (denoted with $\mathcal{E}(j)$) aligns to a subvector (or a "window") of each haplotype whose length is at most ℓ, where ℓ is the maximum length of a fragment.

In the following, a *haplotype window* is an ℓ-long vector over $\{0, 1\}$ and, as usual, given a vector v, $v[i_1 : i_2]$ represents the subvector of v between positions i_1 and i_2 (included and 1-based). We say that a haplotype window $\hat{h}'$ *overlaps* with another haplotype window $\hat{h}$ if $\hat{h}'[2 : \ell] = \hat{h}[1 : \ell - 1]$. We define $D[j, (\hat{h}_1, \ldots, \hat{h}_k)]$ as the minimum number of corrections

needed by fragments in $\bigcup_{t=1}^{j} \mathcal{E}_t$ to reconstruct k haplotypes $(h_1, \ldots, h_k)$ such that, for each h_e, $h_e[j - \ell + 1 : j]$ is equal to $\hat{h}_e$. The algorithm proceeds column-wise computing the value $D[j, (\hat{h}_1, \ldots, \hat{h}_k)]$ for each column p_j and for each collection of k haplotype windows $(\hat{h}_1, \ldots, \hat{h}_k)$ as follows:

$$D[j, (\hat{h}_1, \ldots, \hat{h}_k)] = \Delta(j, (\hat{h}_1, \ldots, \hat{h}_k)) + \min D[j - 1, (\hat{h}'_1, \ldots, \hat{h}'_k)]$$
$$\text{for each } \hat{h}'_e \text{ overlapping with } \hat{h}_e \text{ with } 1 \leq e \leq k \quad (6)$$

where $\Delta(j, (\hat{h}_1, \ldots, \hat{h}_k))$ is the cost needed to align the fragments in $\mathcal{E}(j)$ to the haplotype windows $\hat{h}_1, \ldots, \hat{h}_k$ and that can be easily computed as follows:

$$\Delta(j, (\hat{h}_1, \ldots, \hat{h}_k)) = \sum_{f_i \in \mathcal{E}(j)} \min_{\hat{h}_e} d_H(\hat{h}_e, f_i[j - \ell + 1 : j]) \quad (7)$$

where d_H is the Hamming distance between the two vectors. Without loss of generality and for the sake of simplicity, we assume that if $j - \ell + 1 \leq 0$, then the expression $d_H(\hat{h}_e, f_i[j - \ell + 1 : j])$ is replaced with $d_H(\hat{h}_e[\ell - j + 1 : \ell], f_i[1 : j])$ in Eq. 7.

The base case of the recurrence is $D[1, (\hat{h}_1, \ldots, \hat{h}_k)] = \Delta(1, (\hat{h}_1, \ldots, \hat{h}_k))$ for each collection of haplotype windows $(\hat{h}_1, \ldots, \hat{h}_k)$. Moreover, the algorithm returns the value $\min_{(\hat{h}_1, \ldots, \hat{h}_k)} D[m, (\hat{h}_1, \ldots, \hat{h}_k)]$ corresponding to the optimum, and a collection of k optimal haplotypes can be reconstructed by backtracking.

The algorithm computes all the values $D[j, (\hat{h}_1, \ldots, \hat{h}_k)]$ for each position j from 1 to m and for each collection of haplotype windows $(\hat{h}_1, \ldots, \hat{h}_k)$. Each haplotype window is an ℓ-long binary vector, hence the number of collections of k haplotype windows is $2^{k\ell}$. As a consequence, $O(m2^{k\ell})$ entries have to be stored for the backtracking phase. Furthermore, given a position j and a collection of haplotype windows $(\hat{h}_1, \ldots, \hat{h}_k)$, each entry $D[j, (\hat{h}_1, \ldots, \hat{h}_k)]$ can be computed by Eq. (6) in time $O^*(2^k)$. Indeed, the number of collections of k haplotype windows overlapping (element-wise) with $(\hat{h}_1, \ldots, \hat{h}_k)$ is 2^k and $\Delta(j, (\hat{h}_1, \ldots, \hat{h}_k))$ can be computed in polynomial time, which is needed to obtain the minimum Hamming distance

between each fragment (there are at most *cov* fragments ending at each position) and a haplotype window (of length ℓ) of the collection. It follows that the total running time is $O^*(2^{k\ell+k})$ and, by storing partial information, it can be decreased to $O^*(2^{k\ell})$.

The following lemma shows the correcteness of the algorithm.

Lemma 16. *Consider a fragment matrix $\mathcal{M}$.*

1. *If $D[j, (\hat{h}_1, \ldots, \hat{h}_k)] = g$, then k haplotypes $(h_1, \ldots, h_k)$ can be reconstructed with g corrections from the fragments in $\bigcup_{t=1}^{j} \mathcal{E}_t$ such that, for each h_e, $h_e[j - \ell + 1 : j]$ is equal to $\hat{h}_e$.*

2. *If k haplotypes $(h_1, \ldots, h_k)$ are reconstructed from the fragments in $\bigcup_{t=1}^{j} \mathcal{E}_t$ with g corrections, then $D[j, (\hat{h}_1, \ldots, \hat{h}_k)] \leq g$ (with $\hat{h}_e = h_e[j-\ell+1 : j]$, for each $e \in \{1, \ldots, k\}$).*

Proof. We prove the lemma by induction on j. Both statements obviously hold for $j = 1$. Assume that lemma holds for $j - 1$, we show that both statements hold for j.

(1) By Eq. (6) there exists a collection of haplotype windows $(\hat{h}'_1, \ldots, \hat{h}'_k)$ overlapping (element-wise) with $(\hat{h}_1, \ldots, \hat{h}_k)$ such that $D[j, (\hat{h}_1, \ldots, \hat{h}_k)] = \Delta(j, (\hat{h}_1, \ldots, \hat{h}_k)) + D[j - 1, (\hat{h}'_1, \ldots, \hat{h}'_k)]$. By induction, there exists k haplotypes $(h'_1, \ldots, h'_k)$ that can be reconstructed with $D[j - 1, (\hat{h}'_1, \ldots, \hat{h}'_k)]$ corrections from the fragments in $\bigcup_{t=1}^{j-1} \mathcal{E}_t$ such that, for each h'_e, $h'_e[j-\ell : j-1]$ is equal to $\hat{h}'_e$. As a consequence, $h'_e[j-\ell+1 : j-1]$ is equal to $\hat{h}_e[1 : \ell-1]$ for each e in $\{1, \ldots, k\}$. Let $(h_1, \ldots, h_k)$ be the collection of j-long haplotypes such that $h_e[1 : j - 1] = h'_e$ and $h_e[j] = \hat{h}_e[\ell]$, for each e in $\{1, \ldots, k\}$. Each fragment in $\mathcal{E}(j)$ aligns to one of $(h_1, \ldots, h_k)$ with $\Delta(j, (\hat{h}_1, \ldots, \hat{h}_k))$ total mismatches while, by induction, the fragments in $\bigcup_{t=1}^{j-1} \mathcal{E}_t$ align with $D[j-1, (\hat{h}'_1, \ldots, \hat{h}'_k)]$ total mismatches. Hence, $(h_1, \ldots, h_k)$ is a solution of k-ploid MEC on the fragments $\bigcup_{t=1}^{j} \mathcal{E}_t$ with cost $g = \Delta(j, (\hat{h}_1, \ldots, \hat{h}_k)) + D[j-1, (\hat{h}'_1, \ldots, \hat{h}'_k)]$.

(2) Let $(\hat{h}'_1, \ldots, \hat{h}'_k)$ and $(\hat{h}_1, \ldots, \hat{h}_k)$ be the collections of haplotype windows such that $\hat{h}'_e$ is equal to $h_e[j-\ell : j-1]$ and $\hat{h}_e$ is equal to $h_e[j-\ell+1 : j]$ for each e in $\{1, \ldots, k\}$. We assume

that g' is the number of corrections needed by fragments in $\bigcup_{t=1}^{j-1} \mathcal{E}_t$ to reconstruct the haplotypes $(h_1[1:j-1], \ldots, h_k[1:j-1])$. By induction, it follows that $D[j-1, (\hat{h}'_1, \ldots, \hat{h}'_k)] \leq g'$. By construction, $\Delta(j, (\hat{h}_1, \ldots, \hat{h}_k))$ is the minimum number of corrections that fragments in $\mathcal{E}(j)$ need for reconstructing the haplotype windows $(\hat{h}_1, \ldots, \hat{h}_k)$. Hence, $g \geq \Delta(j, (\hat{h}_1, \ldots, \hat{h}_k)) + g' \geq \Delta(j, (\hat{h}_1, \ldots, \hat{h}_k)) + D[j-1, (\hat{h}'_1, \ldots, \hat{h}'_k)] \geq D[j, (\hat{h}_1, \ldots, \hat{h}_k)]$. □

From the correctness of the algorithm, it directly follows that:

Theorem 17. *k-ploid MEC is in* FPT *when parameterized by fragment length and number of haplotypes.*

7 Conclusions

Minimum Error Correction is a prominent combinatorial problem for haplotype assembly. Investigating the approximation complexity and the fixed-parameter tractability of MEC has proven useful to develop practical haplotype assembly tools (Bansal and Bafna, 2008; Patterson *et al.*, 2014; He *et al.*, 2010; Pirola *et al.*, 2015b). Despite in this paper we addressed some issues that were left open, some other theoretical questions still need an answer.

In this work, we showed that, under the Unique Games Conjecture, MEC is not approximable within any constant factor. However, the approximation complexity of Gapless MEC and the computational complexity of Binary MEC are still unknown. It would be interesting to explore whether Lemma 1, that we used in this paper for achieving a direct 2-approximation algorithm for Binary MEC and an FPT algorithm for Gapless MEC, is also useful for answering to these open questions.

In Section 6.2, we presented a fixed-parameter algorithm for k-ploid MEC when parameterized by fragment length ℓ and the number k of haplotypes. If applied on diploid MEC, it has a worse time complexity than that specifically presented for the diploid case (Section 4). Unfortunately, the algorithm for diploid MEC relies on Lemma 8, that cannot be easily extended to the k-ploid case. For this reason, another interesting research direction is to investigate whether a novel definition of accordance can be proposed in order to extend both Lemma 1 and the characterization of conflict free fragment matrices given by Lemma 8 to the k-ploid case and, hence, to derive a parameterized algorithm based on such a characterization.

Recent advances in sequencing technologies are radically changing the characteristics of the produced data. For example, long gapless reads with sequencing errors uniformly

distributed will likely be common in the near future. The design of FPT algorithms that exploit these characteristics has been recently started by Pirola *et al.* (2015b) but further improvements (either of the underlying models or of the algorithm's time complexity) will be essential to face the growing availability of these data. Furthermore, the drop in sequencing costs allows large-scale studies of rare diseases. In fact, they are usually caused by rare mutations that can only be reliably discovered by sequencing several related individuals. Hence, we expect an increasing interest in the study of new formulations extending MEC on structured populations (where additional constraints induced by the Mendelian laws of inheritance improve the accuracy of the reconstructed haplotypes (Pirola *et al.*, 2012)), as initially done in (Halldórsson *et al.*, 2011).

Acknowledgements. This work has been stimulated by discussions between PB, GK, and NP during the No.045 NII Shonan workshop on Exact Algorithms for Bioinformatics Research, March 2014, Japan.

Funding. The authors acknowledge the support of the MIUR PRIN 2010-2011 grant 2010LYA9RH (Automi e Linguaggi Formali: Aspetti Matematici e Applicativi), of the Cariplo Foundation grant 2013-0955 (Modulation of anti cancer immune response by regulatory non-coding RNAs), of the FA 2013 grant (Metodi algoritmici e modelli: aspetti teorici e applicazioni in bioinformatica).

References

Aguiar, D. and Istrail, S., 2012. HapCompass: A fast cycle basis algorithm for accurate haplotype assembly of sequence data. *J. of Computational Biology* 19, 577–590.

Aguiar, D. and Istrail, S., 2013. Haplotype assembly in polyploid genomes and identical by descent shared tracts. *Bioinformatics* 29, 352–360.

Aldinucci, M., Bracciali, A., Marschall, T., *et al.*, 2014. High-performance haplotype assembly. *In Computational Intelligence Methods for Bioinformatics and Biostatistics - 11th International Meeting, CIBB 2014, Cambridge, UK, June 26-28, 2014, Revised Selected Papers*, 245–258.

Ausiello, G., Crescenzi, P., Gambosi, G., *et al.*, 1999. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media.

Bansal, V. and Bafna, V., 2008. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics* 24, i153–i159.

Bao, E., Jiang, T., and Girke, T., 2013. BRANCH: boosting RNA-Seq assemblies with partial or related genomic sequences. *Bioinformatics* 29, 1250–1259.

Beerenwinkel, N., Beretta, S., Bonizzoni, P., *et al.*, 2014. Covering pairs in directed acyclic graphs. *In Language and Automata Theory and Applications (LATA)*, vol. 8370 of *LNCS*, 126–137. Springer.

Beerenwinkel, N., Beretta, S., Bonizzoni, P., *et al.*, 2015. Covering pairs in directed acyclic graphs. *The Computer Journal* 58, 1673–1686.

Berger, E., Yorukoglu, D., Peng, J., *et al.*, 2014. Haptree: A novel bayesian framework for single individual polyplotyping using ngs data. *PLoS Comput Biol* 10.

Bonizzoni, P., Della Vedova, G., Dondi, R., *et al.*, 2003. The haplotyping problem: An overview of computational models and solutions. *J. Comput. Sci. Technol.* 18, 675–688.

Bonizzoni, P., Dondi, R., Klau, G.W., *et al.*, 2015. On the fixed parameter tractability and approximability of the minimum error correction problem. *In 26th Annual Symposium on Combinatorial Pattern Matching (CPM)*, vol. 9133 of *LNCS*, 100–113.

Browning, B. and Browning, S., 2008. Haplotypic analysis of Wellcome Trust case control consortium data. *Human Genetics* 123, 273–280.

Chen, Z.Z., Deng, F., and Wang, L., 2013. Exact algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* 29, 1938–45.

Cilibrasi, R., Van Iersel, L., Kelk, S., *et al.*, 2007. The complexity of the single individual SNP haplotyping problem. *Algorithmica* 49, 13–36.

Das, S. and Vikalo, H., 2015. SDhaP: haplotype assembly for diploids and polyploids via semi-definite programming. *BMC Genomics* 16, 260.

Dondi, R., 2012. New results for the Longest Haplotype Reconstruction problem. *Discrete Applied Mathematics* 160, 1299–1310.

Downey, R.G. and Fellows, M.R., 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Duitama, J. *et al.*, 2012. Fosmid-based whole genome haplotyping of a HapMap trio child: evaluation of single individual haplotyping techniques. *Nucleic Acids Research* 40, 2041–2053.

Eriksson, N., Pachter, L., Mitsuya, Y., *et al.*, 2008. Viral population estimation using pyrosequencing. *PLoS Comput Biol* 4, e1000074.

Fouilhoux, P. and Mahjoub, A., 2012. Solving VLSI design and DNA sequencing problems using bipartization of graphs. *Computational Optimization and Applications* 51, 749–781.

Fulkerson, D.R., 1956. Note on Dilworth's decomposition theorem for partially ordered sets. *Proc. American Mathematical Society* 7, 701–702.

Garey, M.R. and Johnson, D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Garg, N., Vazirani, V.V., and Yannakakis, M., 1996. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing* 25, 235–251.

Geraci, F., 2010. A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem. *Bioinformatics* 26, 2217–2225.

Guo, J., Gramm, J., Hüffner, F., *et al.*, 2006. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.* 72, 1386–1396.

Halldórsson, B.V., Aguiar, D., and Istrail, S., 2011. Haplotype phasing by multi-assembly of shared haplotypes: Phase-dependent interactions between rare variants. *In PSB*, 88–99. World Scientific Publishing.

He, D., Choi, A., Pipatsrisawat, K., *et al.*, 2010. Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* 26, i183–i190.

Jiao, Y., Xu, J., and Li, M., 2004. On the k-closest substring and k-consensus pattern problems. *In CPM*, vol. 3109 of *LNCS*, 130–144.

Khot, S., 2002. On the power of unique 2-prover 1-round games. *In STOC*, 767–775. ACM.

Kleinberg, J., Papadimitriou, C., and Raghavan, P., 1998. Segmentation problems. *In STOC*, 473–482. ACM.

Lancia, G., Bafna, V., Istrail, S., *et al.*, 2001. SNPs problems, complexity, and algorithms. *In ESA*, vol. 2161 of *LNCS*, 182–193.

Lippert, R., Schwartz, R., Lancia, G., *et al.*, 2002. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in Bioinformatics* 3, 23–31.

Meidanis, J.a., Porto, O., and Telles, G.P., 1998. On the consecutive ones property. *Discrete Applied Mathematics* 88, 325–354.

Ostrovsky, R. and Rabani, Y., 2002. Polynomial-time approximation schemes for geometric min-sum median clustering. *J. ACM* 49, 139–156.

Patterson, M., Marschall, T., Pisanti, N., *et al.*, 2014. WhatsHap: Haplotype assembly for future-generation sequencing reads. *In RECOMB*, vol. 8394 of *LNCS*, 237–249.

Patterson, M., Marschall, T., Pisanti, N., *et al.*, 2015. WhatsHap: Weighted haplotype assembly for future-generation sequencing reads. *J. of Computational Biology* 6, 498–509.

Pirola, Y., Bonizzoni, P., and Jiang, T., 2012. An efficient algorithm for haplotype inference on pedigrees with recombinations and mutations. *IEEE/ACM Trans. Comput. Biology Bioinform.* 9, 12–25.

Pirola, Y., Della Vedova, G., Bonizzoni, P., *et al.*, 2013. Haplotype-based prediction of gene alleles using pedigrees and SNP genotypes. *In ACM-BCB*, 33–41.

Pirola, Y., Zaccaria, S., Dondi, R., *et al.*, 2015a. HapCol. `http://hapcol.algolab.eu/`.

Pirola, Y., Zaccaria, S., Dondi, R., *et al.*, 2015b. HapCol: accurate and memory-efficient haplotype assembly from long reads. *Bioinformatics* .

Reed, B., Smith, K., and Vetta, A., 2004. Finding odd cycle transversals. *Oper. Res. Lett.* 32, 299–301.

Rizzi, R., Tomescu, A., and Mäkinen, V., 2014. On the complexity of minimum path cover with subpath constraints for multi-assembly. *BMC Bioinformatics* 15, S5.

Song, L. and Florea, L., 2013. CLASS: constrained transcript assembly of RNA-seq reads. *BMC Bioinformatics* 14, 1–8.

Trapnell, C., Williams, B.A., Pertea, G., *et al.*, 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* 28, 516–520.

Wang, R.S., Wu, L.Y., Li, Z.P., *et al.*, 2005. Haplotype reconstruction from snp fragments by minimum error correction. *Bioinformatics* 21, 2456–2462.

Yannakakis, M., 1978. Node-and edge-deletion NP-complete problems. *In Proc. of Symp. Theory of Computing (STOC)*, 253–264. ACM.

	p_1	p_2	p_3	p_4
f_1	1	0	-	1
f_2	-	1	0	0
f_3	0	-	1	1

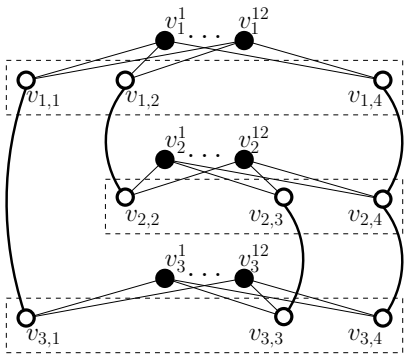

Figure 1: A 3×4 fragment matrix (left) and the associated *fragment graph* (right). Fragment-nodes are in black, while entry-nodes are in white.

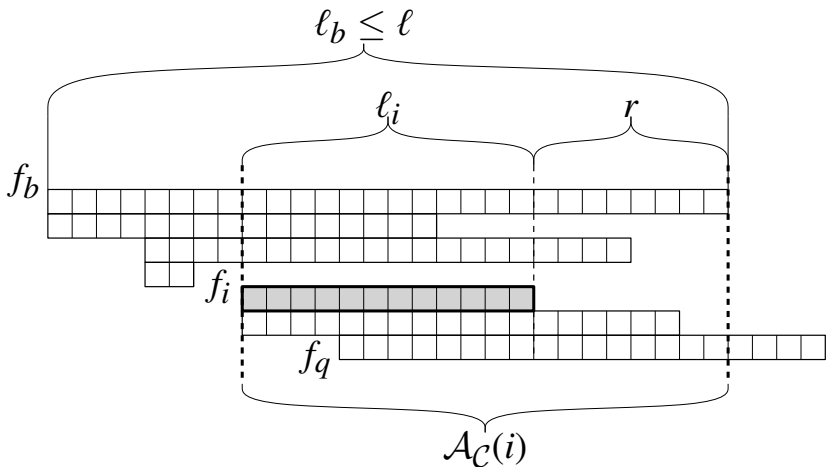

Figure 2: The set $\mathcal{A}_C(f_i)$ of active columns for a fragment f_i. There are r columns p_j to the right of f_i such that there exist two fragments f_b, f_q with $b < i < q$ and $p_j[b], p_j[q] \neq -$. Since fragments are sorted by starting position, we have that $|\mathcal{A}_C(f_i)| = \ell_i + r \leq \ell_b \leq \ell$.

Table 1: Current knowledge of computational complexity, approximability, and fixed-parameter tractability for MEC and its variants, Gapless MEC and Binary MEC. Notice that the expression "all-het" states that the corresponding result holds only under the all-heterozygous assumption, while UCG is the Unique Games Conjecture by Khot (2002). The results on parameterized complexity hold for MEC, hence the negative result holds only for MEC while the positive results also hold for its restrictions.

	Computational complexity	*Approximability*	*Parameterized complexity*
MEC	NP-hard (Lippert *et al.*, 2002)	APX-hard (Cilibrasi *et al.*, 2007) $\notin$ APX under UGC $O(\log nm)$ approxim. (Sect. 3)	$\notin$ XP on c_p and c_f (only MEC, Sect. 3) FPT by ℓ (all-het) (He *et al.*, 2010)
Gapless MEC	NP-hard (Cilibrasi *et al.*, 2007)	?	FPT by *cov* (Patterson *et al.*, 2014)
Binary MEC	?	PTAS (Ostrovsky and Rabani, 2002; Jiao *et al.*, 2004) Simple direct 2-approx (Sect. 5)	FPT by ℓ (Sect. 6.2) FPT by h (Sect. 3)

n number of fragments; m number of SNPs/columns;
ℓ maximum fragment length; *cov* maximum coverage;
h minimum number of corrections; c_p/c_f maximum number of non-hole elements on each column/fragment;

Table 2: Current knowledge of computational complexity, approximability, and fixed-parameter tractability for the newly introduced k-ploid MEC and its variants, k-ploid Gapless MEC and k-ploid Binary MEC. The results on parameterized complexity hold for k-ploid MEC, hence the negative result holds only for k-ploid MEC while the positive results also hold for its restrictions.

	Computational complexity	*Approximability*	*Parameterized complexity*
k-ploid MEC	NP-hard when $k = 2$ (Sect. 6)	$\notin$ APX when $k = 2$ under UGC (Sect. 6)	$\notin$ XP on c_p, c_f, and k (only k-ploid MEC, Sect. 6)
k-ploid Gapless MEC	NP-hard when $k = 2$ (Sect. 6)	?	FPT by *cov* and k (Sect. 6.1)
k-ploid Binary MEC	NP-hard (Cilibrasi *et al.*, 2007)	PTAS (Ostrovsky and Rabani, 2002; Jiao *et al.*, 2004)	FPT by ℓ and k (Sect. 6.2)

ℓ maximum fragment length; *cov* maximum coverage;
c_p/c_f maximum number of non-hole elements on each column/fragment;