BMC Bioinformatics

PROCEEDINGS Open Access

Reconstructing genome mixtures from partial adjacencies

Ahmad Mahmoody*, Crystal L Kahn, Benjamin J Raphael*

From Tenth Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics
Niteroi, Brazil. 17-19 October 2012

Abstract

Many cancer genome sequencing efforts are underway with the goal of identifying the somatic mutations that drive cancer progression. A major difficulty in these studies is that tumors are typically heterogeneous, with individual cells in a tumor having different complements of somatic mutations. However, nearly all DNA sequencing technologies sequence DNA from multiple cells, thus resulting in measurement of mutations from a mixture of genomes. Genome rearrangements are a major class of somatic mutations in many tumors, and the novel adjacencies (i.e. breakpoints) resulting from these rearrangements are readily detected from DNA sequencing reads. However, the assignment of each rearrangement, or adjacency, to an individual cancer genome in the mixture is not known. Moreover, the quantity of DNA sequence reads may be insufficient to measure all rearrangements in all genomes in the tumor. Motivated by this application, we formulate the k-minimum completion problem (*k*-MCP). In this problem, we aim to reconstruct *k* genomes derived from a single reference genome, given partial information about the adjacencies present in the mixture of these genomes. We show that the 1-MCP is solvable in linear time in the cases where: (i) the measured, incomplete genome has a single circular or linear chromosome; (ii) there are no restrictions on the chromosomal content of the measured, incomplete genome. We also show that the *k*-MCP problem, for $k \geq 3$ in general, and the 2-MCP problem with the double-cut-and-join (DCJ) distance are NP-complete, when there are no restriction on the chromosomal structure of the measured, incomplete genome. These results lay the foundation for future algorithmic studies of the *k*-MCP and the application of these algorithms to real cancer sequencing data.

Introduction

Nearly all current genome sequencing studies sequence the DNA from a population of cells rather than from single cells. This is because present DNA sequencing technologies cannot sequence the DNA in a single cell without bias-inducing DNA amplification steps. In the majority of applications, sequencing such a population of cells is not problematic because the DNA in every cell is nearly identical. However, there are two notable examples: metagenomics (e.g. environmental sequencing or microbiome studies) and cancer sequencing. In the former case, the genomic differences between cells are due to the presence of mixtures of microorganisms in the sample. In the latter case, the genomic differences between cells are due to somatic mutations that accumulate in individual tumor cells during the progression of cancer [1].

In this paper, we formulate the problem of inferring the organization of each genome present in a mixture in the case where: (1) the individual genomes result from an unknown sequence of genome rearrangements from a known (reference) genome; (2) the adjacencies (breakpoints) of the genomes in the mixture are measured. This situation arises in cancer genome studies where somatic structural aberrations (including inversions, translocations, duplications, deletions, or other rearrangements of large pieces of DNA) induce novel adjacencies, also called breakpoints, that join in the cancer genome two noncontiguous nucleotides from the normal genome. In current

* Correspondence: ahmad@cs.brown.edu; braphael@cs.brown.edu
Department of Computer Science, Brown University, Providence (RI), USA

cancer sequencing projects, these novel adjacencies are determined from alignments of paired-end reads from cancer DNA to the reference human genome [2,3]. However, these approaches generally do not measure all adjacencies present in the tumor. For example, the quantity of DNA sequence reads (coverage) may be insufficient to measure all adjacencies in all genomes in the tumor, particularly adjacencies that are present in a minority of cancer cells. Moreover, alignment of reads to repetitive regions is challenging, particularly for short reads produced by current sequencing technologies, and thus some adjacencies may not be reliably measured.

We formulate the *k*-Minimum Completion Problem (*k*-MCP) of determining the *k* genomes present in a mixture from a set of measured adjacencies that minimize the total distance between the reference genome and the *k* measured (i.e. cancer) genomes. The *k*-MCP is a general problem that encompasses different subproblems that depend on the genomic distance used and the desired chromosomal content of the measured genomes. We show the following results: (1) A linear time algorithm for the 1-MCP in the double cut and join (DCJ) distance [4] when the desired genome has no restrictions on its chromosomal content; (2) A linear time algorithm for the 1-MCP in the DCJ distance when the desired genome has a single circular or linear chromosome; (3) the *k*-MCP is NP-complete for any distance when $k \geq 3$; and (4) the 2-MCP with DCJ distance is NP-complete when the desired genome has no restrictions on its chromosomal content, or when the desired genome has all circular chromosomes.

We emphasize that the *k*-MCP does not model all the issues arising in cancer sequencing: in particular, we restrict attention to copy-neutral structural variants, and ignore single nucleotide mutations, small indels, or other large copy number aberrations. Single nucleotide mutations and small indels can be addressed separately as they do not produce novel adjacencies of the type studied in *k*-MCP. Copy number aberrations are common in cancer, but appropriate handling of these mutations when measured in a heterogeneous mixture introduces an entirely different set of challenges: e.g. a deletion of a genomic segment in half of the cells in the mixture with a duplication of the same segment in the other half of the cells will be difficult to distinguish from no copy number change. Finally, we assume that all measured adjacencies are real, while in fact there are likely to be false positive adjacencies. Extending the model to consider these additional complexities is left for future work.

In following sections, we first provide a precise formulation of the *k*-MCP and describe related work. Then, we provide algorithms and proofs of the complexity of the problem in various cases.

Definitions and problem statement

In this section we present some preliminary definitions and give the formal definition of *k*-MCP.

A *gene* g is an oriented sequence of nucleotides, with two extremities: a *head* g_h and a *tail* g_t. An *adjacency* is an unordered pair of gene extremities. A *genome* $\mathcal{G}$ on n genes is a set $A(\mathcal{G})$ of adjacencies such that each of the $2n$ gene extremities in $\mathcal{G}$ is a member of at most one adjacency in $A(\mathcal{G})$. The gene extremities which are not members of any adjacency in $A(\mathcal{G})$ are called *telomeres* of $\mathcal{G}$, and we denote the set of all telomeres by $T(\mathcal{G})$ (Figure 1-a). Through this work, we assume that the genes of a genome are distinct.

The *genome graph* of a genome $\mathcal{G}$ is a graph whose labeled vertices are the gene extremities in $\mathcal{G}$, and whose edge set is $A(\mathcal{G})$. We denote the genome graph of $\mathcal{G}$ by $\mathrm{gr}(\mathcal{G})$. Because each extremity is in at most one adjacency of $A(\mathcal{G})$, the graph $\mathrm{gr}(\mathcal{G})$ is a matching graph (not necessarily perfect). Note that the genome graph is uniquely determined by the genome, and conversely. For convenience, we also define the *augmented genome graph* $\overline{\mathrm{gr}}(\mathcal{G})$ to be the genome graph augmented with additional edges connecting extremities of the same gene, i.e., $\overline{\mathrm{gr}}(\mathcal{G})$ is the graph whose labeled vertices are the gene extremities in G, and whose edge set is $A(\mathcal{G}) \cup \{\{g_h, g_t\} | g \text{ is a gene in } \mathcal{G}\}$.

A *chromosome* of $\mathcal{G}$ is the set of all adjacencies and telomeres of gene extremities in a connected component of the augmented genome graph (Figure 1-b). A chromosome is *linear* (resp. *circular*) if the corresponding connected component is a path (resp. cycle) (Figure 1-b). Note that an adjacency $\{g_h, g_t\}$ represents a circular chromosome with the single gene g. A genome is *circular* or *linear* if all of its chromosomes are circular or linear, and we say it is *mixed* if it has both circular and linear chromosomes. A genome is *uni-chromosomal* if it has only one chromosome, and it is *multi-chromosomal*, otherwise. A *chromosomal condition* is a condition on the number or type of chromosomes in a genome. For example we can describe the structure of a genome by two chromosomal conditions: being (i) uni-chromosomal, and (ii) circular.

As described above a paired-end sequencing experiment provides the adjacencies $A(\mathcal{G})$ of the sequenced genome relative to the genes from a reference genome. However, our knowledge about a genome's adjacencies is typically incomplete. For a set $\mathcal{C}$ of chromosomal conditions, a $\mathcal{C}$ *-partial genome* $\mathcal{G}'$ on n genes is a set of adjacencies $A(\mathcal{G}')$ such that there exists a set $\bar{A}(\mathcal{G}')$ of pairs of gene extremities such that $A(\mathcal{G}') \cup \bar{A}(\mathcal{G}')$ is a genome with chromosomal condition $\mathcal{C}$. When $\mathcal{C}$ is clear in the context we will say partial-genome instead of $\mathcal{C}$ -partial genome. The problems we study below

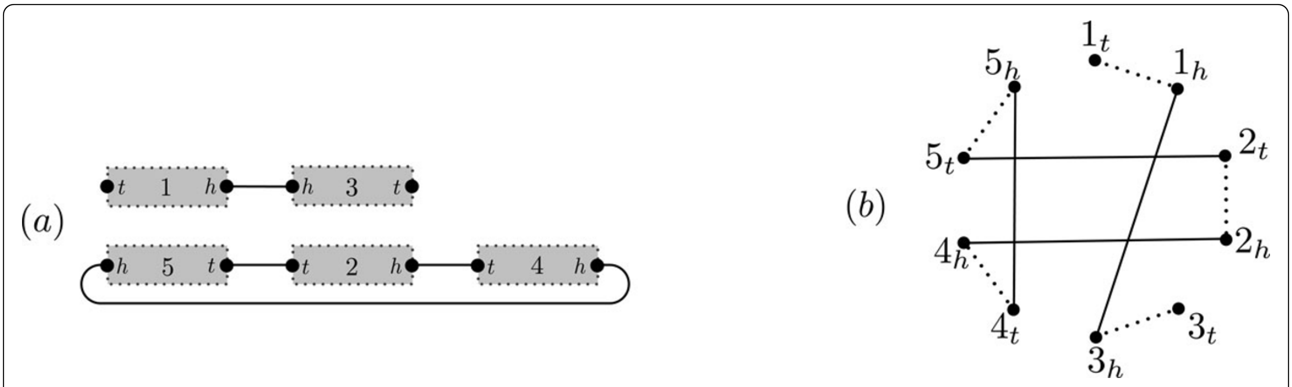

Figure 1 Genome and genome graph. (a) A genome $\mathcal{G}$ on the set of genes {1, 2, 3, 4, 5} with two chromosomes (one linear and one circular). $\mathcal{A}(\mathcal{G}) = \{\{1_h, 3_h\}, \{5_t, 2_t\}, \{2_h, 4_t\}, \{4_h, 5_h\}\}, T(\mathcal{G}) = \{1_t, 3_t\}$. (b) The genome graph (black edges) of $\mathcal{G}$ with additional edges (dotted) connecting the extremities of the same gene. There is one cycle component and one path component.

involve adding the missing adjacencies in $\mathcal{C}$ -partial genomes to complete them into genomes with chromosomal condition $\mathcal{C}$. Sometimes we have an idea about the number or the structure of chromosomes in a genome. We define a *completion* of a partial genome relative to these chromosomal conditions. If $\mathcal{G}$ is a genome, we say $\mathcal{G}' \subseteq \mathcal{G}$ provided $A(\mathcal{G}') \subseteq A(\mathcal{G})$. A *completion*$_\mathcal{C}$ of a partial genome $\mathcal{G}'$ is a genome $\mathcal{G}$ with $\mathcal{G}' \subseteq \mathcal{G}$ and satisfying the conditions in $\mathcal{C}$. When $\mathcal{C}$ is clear in the context, we just say completion instead of completion$_\mathcal{C}$.

A *multi-genome* is a mixture of genomes with the same set of genes. Formally, the multi-genome $\mathcal{M}$ formed from genomes $\mathcal{G}_1, ..., \mathcal{G}_m$ is a multiset $A(\mathcal{M})$ obtained from $A(\mathcal{M}) = \sqcup_{i=1}^m A(\mathcal{G}_i)$, the disjoint union of $A(\mathcal{G}_i)'s$ (For a multiset S and an element *r*, if $c_S(r)$ is the number of copies of *r* in *S*, the disjoint union of two multisets $A \sqcup B$ is a multiset in which each element *r* appears $c_A(r) + c_B(r)$ times.). Note that the partition of the adjacencies in $A(\mathcal{M})$ into $A(\mathcal{G}_1), ..., A(\mathcal{G}_m)$ is not known. There is a corresponding *genome graph*, a *multigraph* whose vertices are the gene extremities, and whose edge set is the multiset $A(\mathcal{M})$. We denote the genome graph of a multi-genome $\mathcal{M}$ by $\text{gr}(\mathcal{M})$.

The genome graph is related to the *breakpoint graph* in genome rearrangement studies. The breakpoint graph $B(\mathcal{G}_1, \ldots, \mathcal{G}_m)$ of the genomes $\mathcal{G}_1, ..., \mathcal{G}_m$ is an edge-colored multigraph whose labeled vertices are the $2n$ gene extremities and whose edges are all the adjacencies in $\sqcup_{i=1}^m A(\mathcal{G}_i)$, with each edge assigned a color according to its genome of origin. Thus, the only difference between the breakpoint graph and the genome graph is the lack of edge-coloring in the latter, reflecting our inability to measure the origin of each adjacency.

Our knowledge about a multi-genome can be incomplete. For example a tumor is a mixture of different cancer genomes, and during sequencing process, we obtain a *mixture* of adjacencies from these genomes. We represent the mixtures of adjacencies by a *partial multi-genome*. A partial multi-genome is a multi-set $\sqcup_{i=1}^m A(\mathcal{G}_i')$, where each $\mathcal{G}_i'$ is partial genome. We define the *genome graph* of a partial multi-genome analogously to a multi-genome.

If *k* is a positive integer and $\mathcal{M}$ is a partial multi-genome, a *k-completion* of $\mathcal{M}$ is a family of *k* genomes $\mathcal{M}^k = \{\mathcal{G}_1, ..., \mathcal{G}_k\}$, such that $\mathcal{M} \subseteq \sqcup_{i=1}^k \mathcal{G}_i$. Note that *existence* of a completion for a partial (multi-) genome is dependent on the structure of the partial (multi-) genome and the chromosomal conditions. Also, the existence of a completion does not imply its uniqueness.

We use a distance function to distinguish between different completions. A *distance* function on pairs of genomes (with the same set of genes), is a measure of dissimilarity between the genomes. Having selected a pairwise distance function we must define a distance between the *k* genomes in a mixture. Motivated by the fact that the different cancer genomes in a tumor are obtained by somatic genome rearrangements from a healthy genome, we model the evolution of the cancer genomes by a rooted tree in which all the cancer genomes are descendants of the healthy one. Suppose $\mathcal{A}$ represents a healthy genome, and $\mathcal{M}^k$ a mixture of *k* cancer genomes obtained by rearrangements of the genome $\mathcal{A}$. A *mixture tree* $\mathcal{T}_{\mathcal{M}^k,\mathcal{A}}$ is a rooted tree on $\mathcal{M}^k \cup \{\mathcal{A}\}$, such that the root vertex is $\mathcal{A}$ and *k* genomes in $\mathcal{M}^k$ are (some of) the vertices in $\mathcal{T}_{\mathcal{M}^k,\mathcal{A}}$. If φ is a distance function on a pair of genomes, then the φ-value of $\mathcal{T}_{\mathcal{M}^k,\mathcal{A}}$, denoted by $\phi(\mathcal{T}_{\mathcal{M}^k,\mathcal{A}})$ is defined as follows:

$$\phi(\mathcal{T}_{\mathcal{M}^k,\mathcal{A}}) = \sum_{\{u,v\} \in E} \phi(u, v),$$

where E is the set of edges in $\mathcal{T}_{\mathcal{M}^k,\mathcal{A}}$.

We now define the k-Minimum Completion Problem.

***k*-Minimum Completion Problem (*k*-MCP)** Given a $\mathcal{C}$ -partial multi-genome $\mathcal{M}$, a positive integer k, a reference genome $\mathcal{A}$, and a distance function φ, find a k-completion $\mathcal{M}^k$ and a mixture tree $\mathcal{T}_{\mathcal{M}^k,\mathcal{A}}$ such that $\phi(\mathcal{T}_{\mathcal{M}^k,\mathcal{A}})$ is minimum over all k-completions and mixture trees. If no k-completion exists for $\mathcal{M}$, we say that this k-MCP does not have a *valid* solution. We say the k-MCP is *unrestricted* if $\mathcal{C} = \emptyset$, and is *restricted*, otherwise.

As written, the k-MCP is a general problem that encompasses many subproblems depending on chromosomal condition set $\mathcal{C}$ and the distance φ. Common distances in genome rearrangement studies include the *breakpoint distance* [5], the *Hannenhalli-Pevzner* distance [6] (which generalizes the *reversal distance* [7]), and the *double-cut-and-join (DCJ) distance* [4]. Below we will use the DCJ distance, which approximates the other distances [8].

For two genomes $\mathcal{G}_1$ and $\mathcal{G}_2$ on the same set of n genes, their *double-cut-and-join (DCJ)* distance, denoted by $d_{DCJ}(\mathcal{G}_1, \mathcal{G}_2)$, is equal to

$$n - c(\mathcal{G}_1, \mathcal{G}_2) - \frac{p(\mathcal{G}_1, \mathcal{G}_2)}{2},$$

where $c(\mathcal{G}_1, \mathcal{G}_2)$ is the number of cycles in $B = B(\mathcal{G}_1, \mathcal{G}_2)$ and $p(\mathcal{G}_1, \mathcal{G}_2)$ is the number of paths in B with odd number of vertices [8].

Remark. When at least one of the $\mathcal{G}_i's$ are circular we have $p(\mathcal{G}_1, \mathcal{G}_2) = 0$ and d_{DCJ} (G_1, G_2) = n - c. Thus, having a larger number of cycles in their breakpoint graph is equivalent to having a smaller distance.

Related work

In comparison to other genome rearrangement problems considered in the literature, the k-MCP has three distinguishing features. (1) The input is a mixture of adjacencies from multiple genomes and the genome of origin of each adjacency is unknown. (2) The set of adjacencies is incomplete: not every adjacency from every genome in the mixture is measured. (3) The ancestral relationships between the genomes in the mixture are unknown, and might include both "ancestral" and "present day" genomes. Some of these features have been considered individually in other work, but to our knowledge no previous work has considered all three together. The first feature bears some resemblance to the genome halving problem [9] of finding the doubled ancestor genome by minimizing a rearrangement distance. This problem and further generalizations to polyploidization [10] involves partitioning (or coloring) adjacencies to minimize a rearrangement distance. However, in general no adjacencies are missing and the distance is pairwise (i.e., no tree) in contrast to the 2-MCP.

Regarding the second feature, several authors have considered the problem of inferring missing adjacencies in a manner that optimizes a genome rearrangement distance. Notably, [11] and [12] consider the problem of computing reversal distance between pairs of partially assembled genomes that are provided as unordered sequences of contigs. These problems were motivated by limitations in DNA sequence technologies that result in most whole-genome assemblies being highly fragmented and comprised of contigs whose relative ordering is unknown. These problems are variations of the 1-MCP, where the reference genome $\mathcal{A}$ also has missing adjacencies. In particular, [12] orient sets of contigs from two genomes in such a way that the number of cycles in the breakpoint graph of the resulting genomes is maximized, which they note "has been shown to approximate very well the reversal distance between them." However, there is no work on extending this analysis to more than two genomes.

Regarding the third feature, the genome median problem considers the problem of finding an ancestral genome that minimizes the distance between three given genomes [5,13]. This is different from k-MCP in that the three individual genomes are known (rather than mixed) and the genomes are complete with no missing adjacencies. Also, in the median problem the topology of the phylogenetic tree has been already inferred, while in k-MCP we have to find an optimal topology for the phylogenetic tree as well.

Results

In this section we first consider the 1-MCP problem. We present linear time algorithms that solve 1-MCP in the cases where: (i) the measured, incomplete genome has a single circular or linear chromosome; (ii) there are no restrictions on the chromosomal content of the measured, incomplete genome.

Next we prove that the unrestricted k-MCP is NP-complete when $k \geq 3$ for any distance function φ. Finally, we show that the unrestricted 2-MCP, and the restricted 2-MCP where all chromosomes are circular (i.e., $\mathcal{C}$ = {circular}), are NP-complete for DCJ distance.

1-MCP

Here, we consider the unrestricted 1-MCP and two restricted versions of 1-MCP problem: (1) the chromosomal condition set is {*circular*, uni-chromosomal}, which we denote by 1-MCP$_c$; (2) the chromosomal condition set is {*linear*, uni-chromosomal}, which we denote by 1-MCP$_\ell$. We first show that unrestricted version is linearly tractable. Then, we show that we can solve the

1-MCP$_c$ in linear time. Finally, we prove a relation between 1-MCP$_c$ and, 1-MCP$_\ell$ which implies that 1-MCP$_\ell$ is also solvable in linear time.

Note that 1-MCP$_\ell$ is a variation of the Block Ordering Problem (BOP) considered in [12]. In our terminology, the BOP considers two partial genomes, and aims to complete both partial genomes into linear, unichromosomal genomes such that the pairwise distance between the completed genome is minimal. In [12], Gaul and Blanchette provide a linear algorithm for BOP. The algorithm we present for 1-MCP$_\ell$ is simpler than the algorithm for the BOP in [12], and our algorithm is obtained from a straightforward algorithm (Algorithm 1 below) which solves 1-MCP$_c$ in linear time.

We begin with the unrestricted 1-MCP, where we have the following result.

Theorem 1. The unrestricted 1-MCP with DCJ distance is linearly tractable.

Proof. In 1-MCP we have a single partial genome $\mathcal{G}$ and a reference genome $\mathcal{A}$ (see Figure 2-a). Since both $\mathcal{G}$ and $\mathcal{A}$ are matchings over the gene extremities, their breakpoint graph $B(\mathcal{G},\mathcal{A})$ consists of some paths and cycles. Suppose $P_1, \ldots, P_r$ are all the paths such that the first and their last edges are adjacencies in $\mathcal{A}$. An optimal completion for $\mathcal{G}$ can be obtained by adding an edge to $\mathcal{G}$ which connects the end points of each P_i, for $1 \le i \le r$ (see Figure 3), since we only can add edges between the vertices which are not incident with any edge in $A(\mathcal{G})$, i.e., the end vertices of P_i's. Note that adding other possible edges just create longer paths in $B(\mathcal{G},\mathcal{A})$. □

1-MCP$_c$: circular uni-chromosomal completion

Here we consider 1-MCP$_c$, the restricted 1-MCP for a partial genome $\mathcal{G}$ that we wish to complete to a circular uni-chromosomal genome $\mathcal{G}_c$. We assume that $\mathcal{G}$ is not already a circular uni-chromosomal genome. Thus $\mathcal{G}$ has a set $F(\mathcal{G})$ of *free* extremities, i.e., the extremities that are not in any adjacency in $\mathcal{G}$. Equivalently, $F(\mathcal{G})$ is the set of vertices of degree 0 in the genome graph $\mathrm{gr}(\mathcal{G})$. Finding the completion $\mathcal{G}_c$ corresponds to finding a partition of $F(\mathcal{G})$ into pairs of extremities, i.e., into adjacencies. However, this partition cannot be arbitrary as the adjacencies defined by the partition must satisfy two constraints: (1) The resulting genome $\mathcal{G}_c$ is circular uni-chromosomal, meaning that the augmented genome graph $\overline{\mathrm{gr}}(\mathcal{G}_c)$ has exactly one component, a cycle. Note that $\overline{\mathrm{gr}}(\mathcal{G})$ has only path components, since $\overline{\mathrm{gr}}(\mathcal{G}) \subset \overline{\mathrm{gr}}(\mathcal{G}_c)$ and $\mathcal{G} \neq \mathcal{G}_c$. (2) The resulting genome $\mathcal{G}_c$ must minimize the distance between the reference genome $\mathcal{A}$ and $\mathcal{G}_c$.

The first constraint on partitioning of $F(\mathcal{G})$ is that joining extremities at ends of a same path in $\overline{\mathrm{gr}}(\mathcal{G})$ by an edge, which we call an *excluded edge*, creates a cycle. This cycle must be selected carefully to obtain a uni-chromosomal genome. We define $E(\mathcal{G})$ to be the set of all excluded edges.

The second constraint on partitioning of $F(\mathcal{G})$ is provided by our desire to minimize the distance between the reference genome $\mathcal{A}$ and $\mathcal{G}_c$. For the DCJ distance, we must maximize the number $c(\mathcal{A},\mathcal{G}_c)$ of cycles in the breakpoint graph $B = B(\mathcal{G},\mathcal{A})$. Adding an edge to $A(\mathcal{G})$ increases the number of cycles in B if and only if the edge connects the endpoints of a same path in B. We call such an edge a *desired* edge and denote by $\mathcal{D}_{\mathcal{A}}(\mathcal{G})$ the set of all desired edges. Now we combine these two constraints into a graph.

We define the *free-extremities graph*, $R = R(\mathcal{G},\mathcal{A})$ to be a bicolored graph, whose vertex set is $F(\mathcal{G})$, and whose edge set is $D_{\mathcal{A}}(\mathcal{G}) \sqcup E(\mathcal{G})$. The edges from $D_{\mathcal{A}}(\mathcal{G})$ are colored *blue* and the edges from $E(\mathcal{G})$ are colored *red*. Note that R is a multi-graph, and R consists of even cycles. This is because both $D_{\mathcal{A}}(\mathcal{G})$ and $E(\mathcal{G})$

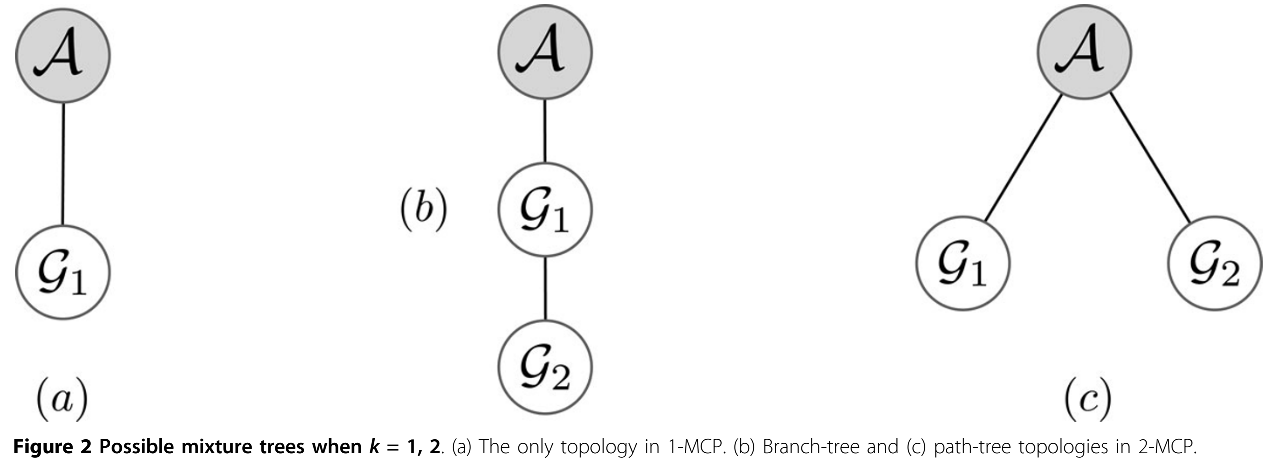

Figure 2 Possible mixture trees when k = 1, 2. (a) The only topology in 1-MCP. (b) Branch-tree and (c) path-tree topologies in 2-MCP.

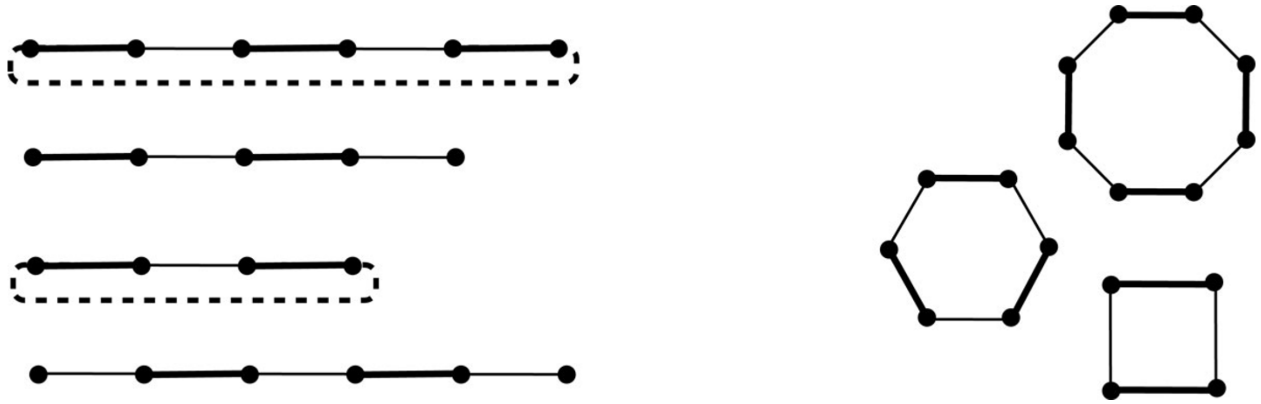

Figure 3 The breakpoint graph $B(\mathcal{G}, \mathcal{A})$ with possible edges to be added to the adjacencies of $\mathcal{G}$. Breakpoint graph $B(\mathcal{G}, \mathcal{A})$ consisting of paths and cycles. Thick edges are in $A(\mathcal{A})$, and thin edges are in $A(\mathcal{G})$. Dashed edges are the edges that should be added to $A(\mathcal{G})$ for the paths whose first and last edges are in $A(\mathcal{A})$.

are perfect matchings on $F(\mathcal{G})$: since both $A(\mathcal{A})$ and $\{\{g_h, g_t\} \mid g$ is a gene in $\mathcal{G}\}$ are perfect matchings on the set of all gene extremities. The restriction of these perfect matchings to $F(\mathcal{G})$ are $D_{\mathcal{A}}(\mathcal{G})$ and $E(\mathcal{G})$. See Figure 4-b. Thus, we have

$$|E(\mathcal{G})| = |D_{\mathcal{A}}(\mathcal{G})| = \frac{|F(\mathcal{G})|}{2}. \qquad (1)$$

To find a completion of the partial genome $\mathcal{G}$ we select pairs $\{u, v\}$ of free extremities from $F(\mathcal{G})$ and add them as adjacencies to $A(\mathcal{G})$. Respecting the constraints encoded in the free-extremities graph R, we define a transformation `update`$(R, \{u, v\})$ that records the effect of adding adjacency $\{u, v\}$ to $\mathcal{G}$ (Figure 4). In particular, since u and v are free vertices of $\mathcal{G}$, there are paths P^u_B and P^v_B in B with an endpoint equal to u and v, respectively. Similarly, there are paths $P^u_{\overline{\mathrm{gr}}(\mathcal{G})}$ and $P^v_{\overline{\mathrm{gr}}(\mathcal{G})}$ in $\overline{\mathrm{gr}}(\mathcal{G})$ having an endpoint equal to u and v, respectively. We may have $P^u_B = P^v_B$ or $P^u_{\overline{\mathrm{gr}}(\mathcal{G})} = P^v_{\overline{\mathrm{gr}}(\mathcal{G})}$. By the definition of $D_{\mathcal{A}}(\mathcal{G})$, P^u_B and P^v_B are represented by blue edges b^u and b^v in R incident to u and v. Similarly by the definition of $E(\mathcal{G})$, $P^u_{\overline{\mathrm{gr}}(\mathcal{G})}$ and $P^v_{\overline{\mathrm{gr}}(\mathcal{G})}$ are represented by red edges r^u and r^v in R incident to u and v. Adding the adjacency $\{u, v\}$ to $A(\mathcal{G})$ will have the following effects on B and $\overline{\mathrm{gr}}(\mathcal{G})$:

(i) u and v are no longer free vertices.

(ii) If $P^u_B \neq P^v_B$ then these paths merge into one path in $B \cup \{u, v\}$. Otherwise these paths merge to create a cycle in $B \cup \{u, v\}$, and the number of cycles in the breakpoint graph increases by one.

(iii) If $P^u_{\overline{\mathrm{gr}}(\mathcal{G})} \neq P^v_{\overline{\mathrm{gr}}(\mathcal{G})}$ these paths merge into one path in $\overline{\mathrm{gr}}(\mathcal{G}) \cup \{u, v\}$. Otherwise these paths merge into a cycle in $\overline{\mathrm{gr}}(\mathcal{G}) \cup \{u, v\}$. In the latter case, we should add $\{u, v\}$ as an adjacency if and only if $F(\mathcal{G}) = \{u, v\}$. This is because adding $\{u, v\}$ creates a cycle component in $\overline{\mathrm{gr}}(\mathcal{G}) \cup \{u, v\}$ (i.e., a circular chromosome) and if there are other free vertices any subsequent completion will *not* be uni-chromosomal.

Therefore, adding the adjacency $\{u, v\}$ to $A(\mathcal{G})$ will have three corresponding effects on R: removing the vertices u and v from R based on (i) above, identifying b^u and b^v based on (ii) above, and identifying r^u and r^v based on (iii) above. We denote this process of updating R by `update`$(R, \{u, v\})$. Figure 4 gives an illustration of this process.

If $\{u, v\}$ is a blue edge in R, then `update`$(R, \{u, v\})$ increases the number of cycles in the breakpoint graph B by one. Hence, to find a solution to 1-MCP$_c$ we want to perform `update`$(R, \{u, v\})$ transformations with as many blue edges as possible. On the other hand, adding new adjacencies has to merge the paths in the graph $\overline{\mathrm{gr}}(\mathcal{G})$ in such a way that we end with a genome with exactly one circular chromosome. Let $M_b(R)$ be the maximum possible number of `update` transformations using blue edges for the graph R. The following theorem provides the exact value of $M_b(R)$.

Theorem 2. Suppose $\mathcal{G}$ is a partial genome, $\mathcal{A}$ is a reference genome, and $R = R(\mathcal{G}, \mathcal{A})$ is their free-extremities graph. We have

$$M_b(R) = N_b(R) - c(R) + 1,$$

where $N_b(R)$ is the number of blue edges, and $c(R)$ is the number of cycles in R.

Proof. We prove the theorem by induction on $N_b(R)$. Suppose $N_b(R) = 1$. Then necessarily R consists of a cycle of length 2 with one blue and one red edge, and c

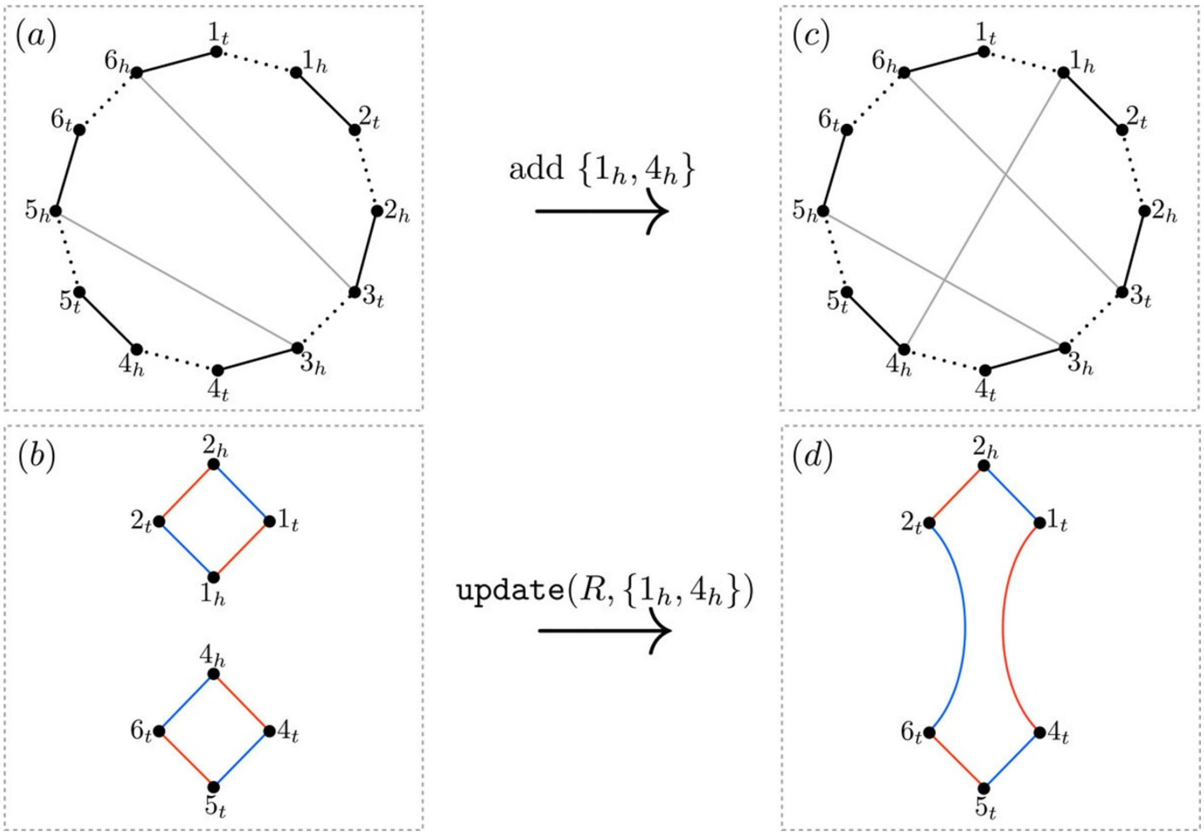

Figure 4 Adding adjacencies to a partial genome $\mathcal{G}$ to solve the 1-MCP$_c$. (a) The breakpoint graph $B(\mathcal{G}, \mathcal{A})$. Gray edges indicate adjacencies of $\mathcal{G}$, black edges indicate adjacencies of $\mathcal{A}$, and the dotted edges connect extremities of the same gene. The set of free vertices is $F(\mathcal{G}) = \{1_t, 1_h, 2_t, 2_h, 4_t, 4_h, 5_t, 6_t\}$. (b) The free-extremities graph $R(\mathcal{G}, \mathcal{A})$ consists of two even cycles. Blue edges are desired edges $D_{\mathcal{A}}(\mathcal{G})$ and red edges are excluded edges $E(\mathcal{G})$. (c) The resulting breakpoint graph after adding adjacency $\{1_h, 4_h\}$. (d) The resulting free-extremities graph after `update`$(R, \{1_h, 4_h\})$. The vertices 1_h and 4_h are no longer free extremities and thus are removed during `update`$(R, \{1_h, 4_h\})$.

(R) = 1. Thus, we update the graph R with the unique (and the only possible) blue edge obtaining

$$M_b(R) = 1 = N_b(R) - c(R) + 1.$$

Now suppose $N_b(R) > 1$. Then $|E(\mathcal{G})| > 1$, since $|E(\mathcal{G})| = |D_{\mathcal{A}}(\mathcal{G})| = N_b$. Suppose $u, v \in F(\mathcal{G})$, and $\{u, v\} \notin E(\mathcal{G})$, i.e., there is no red edge between u and v in R. Then, we have the following three cases for u and v: (i) u and v are from different cycles C_u and C_v in R, (ii) u and v are connected with a blue edge in a cycle C of R, or (iii) u and v are non-neighboring vertices in a cycle C of R.

Let R' = `update`$(R, \{u, v\})$ be the free-extremities graph after the update. Since u and v are incident with blue edges in R, after `update`$(R, \{u, v\})$ the number of blue edges decreases by one, i.e., $N_b(R') = N_b(R) - 1$.

Thus, by induction hypothesis

$$M_b(R') = N_b(R') - c(R') + 1 = N_b(R) - c(R'). \qquad (2)$$

Considering the above cases we have:

(i) After `update`$(R, \{u, v\})$, C_u and C_v will shrink into one cycle, and $c(R') = c(R) - 1$. Thus by (2), $M_b(R') = N_b(R) - c(R) + 1$. By choosing such an edge we can update R with $N_b(R) - c(R) + 1$ blue edges.
(ii) After `update`$(R, \{u, v\})$, C shrinks into a smaller cycle, and $c(R') = c(R)$. Thus, by (2), $M_b(R') = N_b(R) - c(R)$. Since $\{u, v\}$ is a blue edge, we can update R with $N_b(R) - c(R) + 1$ blue edges.
(iii) After `update`$(R, \{u, v\})$, C splits into two smaller cycles. Thus $c(R') = c(R) + 1$. Thus, by (2), $M_b(R') = N_b(R) - c(R) - 1$. So by choosing $\{u, v\}$ we can update R with $N_b(R) - c(R) - 1$ blue edges.

By calculations above, choosing a pair $\{u, v\}$ satisfying cases (i) or (ii) will result in a greater number of `update` moves with blue edges, than choosing a pair satisfies the case (iii). Moreover, considering pairs $\{u, v\}$ from cases (i) and (ii) gives $M_b(R) = N_b(R) - c(R) + 1$. □

We call a pair $\{u, v\}$ (which may or may not be an edge in R) satisfying case (i) or (ii) in the proof of Theorem 2 an *optimal* adjacency. Optimal adjacencies play an important role in finding a solution of 1-MCP$_c$: updating the free-extremities graph with these adjacencies results in the maximum number of blue edges used in `update` transformations. We have the following important corollary to this theorem.

Corollary 1. Suppose $\mathcal{G}$ is a partial genome and $\mathcal{A}$ is a reference genome. Adding any optimal adjacency to $A(\mathcal{G})$ leads to a solution for 1-MCP$_c$. In other words, for any optimal adjacency e, there exists a solution $\mathcal{G}_c$ for 1-MCP$_c$ which includes e as an adjacency.

Proof. By Theorem 2, adding any optimal adjacency to $A(\mathcal{G})$ will allow the maximum number of blue edges in the `update` process. Since each `update` transformation on a blue edge increases the number of cycles in the breakpoint graph by one, a sequence of `update` transformations on optimal adjacencies gives a solution $\mathcal{G}_c$ to 1-MCP$_c$. Hence, if $\mathcal{G}_c$ is the resulting completion of $\mathcal{G}$, we obtain the maximum number of cycles in the breakpoint graph $B(\mathcal{G}_c, \mathcal{A})$. □

A linear time (in number of genes) algorithm for solving 1-MCP$_c$ adds optimal adjacencies according to cases (i) and (ii) in Theorem 2, and is shown in Algorithm 1. The following corollary is an immediate consequence of Corollary 1 and Algorithm 1.

Corollary 2. The 1-MCP$_c$ is solvable in linear time.

Algorithm 1: Solving 1-MCP$_c$

Input : Partial genome $\mathcal{G}$ and reference genome A.

Output: A 1-completion $\mathcal{G}_c$ that is circular uni-chromosomal and maximizes $c(\mathcal{G}_c, \mathcal{A})$.

```
begin
  Construct the free-extremities graph $R = R(\mathcal{G}, \mathcal{A})$;
  $\mathcal{G}_c \leftarrow \mathcal{G}$;
  while $c(R) > 1$ do
    $u, v \leftarrow$ select two vertices from different cycles in $R$;
      $\mathcal{A}(\mathcal{G}_c) \leftarrow \mathcal{A}(\mathcal{G}_c) \cup \{u, v\}$;
      $R \leftarrow$ update $(R, \{u, v\})$;
  while the number of blue edges in $R > 1$ do
      $u, v \leftarrow$ select two vertices connected via a blue edge in $R$;
      $\mathcal{A}(\mathcal{G}_c) \leftarrow \mathcal{A}(\mathcal{G}_c) \cup \{u, v\}$;
      $R \leftarrow$ update $(R, \{u, v\})$;
   Add the single remaining excluded edge in $E(\mathcal{G})$ to $A(\mathcal{G}_c)$;
   Output the resulting circular uni-chromosomal genome $\mathcal{G}_c$;
end
```

1-MCP$_\ell$: linear uni-chromosomal completion

In this section we consider the 1-MCP with chromosomal condition of a linear uni-chromosomal genome. We refer to this restricted problem as 1-MCP$_\ell$. We relate solutions of 1-MCP$_\ell$ to solutions of 1-MCP$_c$. Combined with the results in the previous section, we derive a linear time algorithm for 1-MCP$_\ell$.

Recall that $\hat{c}_c(\mathcal{G}, \mathcal{A})$ is the number of alternating cycles in the breakpoint graph $B(\mathcal{G}_c, \mathcal{A})$, for any solution $\mathcal{G}_c$ of 1-MCP$_c$. Similarly, we define $\hat{c}_\ell(\mathcal{G}, \mathcal{A})$ to be the number of alternating cycles in $B(\mathcal{G}_\ell, \mathcal{A})$, for any solution $\mathcal{G}_\ell$ of 1-MCP$_\ell$. The following theorem relates the solutions of 1-MCP$_c$ to the solutions of 1-MCP$_\ell$.

Theorem 3. Let $\mathcal{G}$ be a partial genome, $\mathcal{A}_c$ be a circular uni-chromosomal genome, and $\mathcal{A}_\ell$ be a linear uni-chromosomal genome obtained from $\mathcal{A}_c$ by removing an adjacency e. Suppose $\mathcal{A}_c$ and $\mathcal{A}_\ell$ are the reference genomes in 1-MCP$_c$ and 1-MCP$_\ell$, respectively. From any solution $\mathcal{G}_c$ to 1-MCP$_c$ we obtain a solution $\mathcal{G}'_\ell$ for 1-MCP$_\ell$. Also, from any solution $\mathcal{G}_\ell$ to 1-MCP$_\ell$ we obtain a solution $\mathcal{G}'_c$ for 1-MCP$_c$. Moreover, $\hat{c}_c(\mathcal{G}, \mathcal{A}_c) = \hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell) + \theta(e)$, where

$$\theta(e) = \begin{cases} 2 \text{ if } e \text{ is in a cycle in } B(\mathcal{G}, \mathcal{A}_c), \\ 1 \text{ otherwise.} \end{cases}$$

Proof. First, suppose e is not in any cycle in the graph $B(\mathcal{G}, \mathcal{A}_c)$, and hence $\theta(e) = 1$. Let $\mathcal{G}_c$ be a solution to 1-MCP$_c$, and let $\mathcal{G}'_\ell$ be a linear uni-chromosomal genome obtained from $\mathcal{G}_c$ by removing an adjacency $f \in A(\mathcal{G}_c)\backslash A(\mathcal{G})$, such that f and e are in the same cycle in $B(\mathcal{G}_c, \mathcal{A}_c)$. Note that such edge f exists, since e is not in any cycle in $B(\mathcal{G}, \mathcal{A}_c)$ but it is in a cycle of $B(\mathcal{G}_c, \mathcal{A}_c)$. See Figure 5. Both $\mathrm{gr}(\mathcal{G}_c)$ and $\mathrm{gr}(\mathcal{A}_c)$ are perfect matchings as $\mathcal{A}_c$ and $\mathcal{G}_c$ are both circular. Removing the edges e and f from $B(\mathcal{G}_c, \mathcal{A}_c)$ will decrease the number of cycles by exactly one since e and f are in a same cycle in $B(\mathcal{G}_c, \mathcal{A}_c)$. Hence $c(\mathcal{G}'_\ell, \mathcal{A}_\ell) = c(\mathcal{G}_c, \mathcal{A}_c) - 1$, and we have,

$$\hat{c}_c(\mathcal{G}, \mathcal{A}_c) - 1 = c(\mathcal{G}'_\ell, \mathcal{A}_\ell) \leq \hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell), \tag{3}$$

where the last inequality follows from the definition of $\hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell)$ as the largest number of cycles in *any* linear chromosomal completion of $\mathcal{G}$.

Now suppose $\mathcal{G}_\ell$ is a solution to 1-MCP$_\ell$, so $|E(\mathcal{G}_\ell)| = |E(\mathcal{A}_\ell)| = 1$. Assume $E(\mathcal{G}_\ell) = \{f'\}$. Let $\mathcal{G}'_c$ be the circular uni-chromosomal genome obtained by adding f' to $\mathcal{G}_\ell$. Note that there is at least one path component in $B(\mathcal{G}_\ell, \mathcal{A}_\ell)$ which becomes a cycle after adding the edges f to $A(\mathcal{G}_\ell)$ and e to $A(\mathcal{A}_\ell)$. Hence, $\hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell) + 1 = c(\mathcal{G}_\ell, \mathcal{A}_\ell) + 1 \leq c(\mathcal{G}'_c, \mathcal{A}_c) \leq \hat{c}_c(\mathcal{G}, \mathcal{A})$, and we have

$$\hat{c}_\ell(\mathcal{G}_\ell, \mathcal{A}_\ell) \leq \hat{c}_c(\mathcal{G}, \mathcal{A}_c) - 1. \tag{4}$$

Thus by (3) and (4) we have $\hat{c}_c(\mathcal{G}, \mathcal{A}_c) = \hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell) + 1$, which implies that $c(\mathcal{G}'_c, \mathcal{A}_c) = \hat{c}_c(\mathcal{G}, \mathcal{A}_c)$ and $c(\mathcal{G}'_\ell, \mathcal{A}_\ell) = \hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell)$. This means that $\mathcal{G}'_c$ and $\mathcal{G}'_\ell$ are solutions to 1-MCP$_c$ and 1-MCP$_\ell$ that are obtained

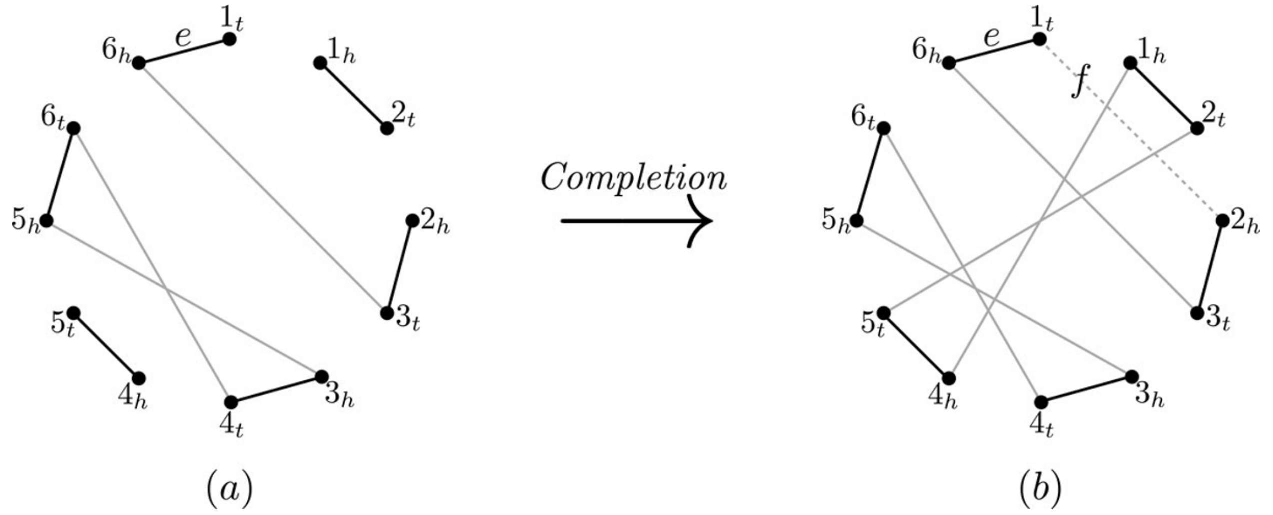

Figure 5 Relating 1-MCP$_c$ and 1-MCP$_\ell$. (a) The breakpoint graph $B = B(\mathcal{G}, \mathcal{A}_c)$; black edges are $A(\mathcal{A}_c)$ and and gray edges are $A(\mathcal{G})$. The edge $e = \{1_t, 6_h\}$ is the only edge in $A(\mathcal{A}_c)\backslash A(\mathcal{A}_\ell)$. Since e is not in a cycle component of B, we have $\theta(e) = 1$. (b) The breakpoint graph $B' = B(\mathcal{G}_c, \mathcal{A}_c)$, where $\mathcal{G}_c$ is a completion of $\mathcal{G}$ and a solution to 1-MCP$_c$. The adjacency f is in $A(\mathcal{G}_c)$ and shown by a gray dashed edge. B' has two cycles, and removing e and f decreases the number of cycles by one.

from $\mathcal{G}_\ell$ and $\mathcal{G}_c$, respectively, which completes the proof for the case $\theta(e) = 1$.

Now suppose e is in a cycle in $B(\mathcal{G}, \mathcal{A}_c)$, and thus $\theta(e) = 2$. Using the same argument above, we have $\hat{c}_c(\mathcal{G}, \mathcal{A}_c) - 2 = c(\mathcal{G}'_\ell, \mathcal{A}_\ell) \leq \hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell)$ since we cannot find such edge f and the number of cycles in $B(\mathcal{G}_c, \mathcal{A}_c)$ decreases by two, when we remove an edge from $\mathcal{G}_c$ (to obtain a linear genome), and e from $\mathcal{A}_c$ (to obtain the genome $\mathcal{A}_\ell$). Also, $\hat{c}_\ell(\mathcal{G}, \mathcal{A}_\ell) + 2 \leq \hat{c}_c(\mathcal{G}, \mathcal{A}_c)$, as adding the excluded edges of $\mathcal{A}_\ell$ and $\mathcal{G}_\ell$ will increase the number of cycles by 2. Thus, for this case we have $\hat{c}_c(\mathcal{G}, \mathcal{A}) = \hat{c}_\ell(\mathcal{G}, \mathcal{A}) + 2$ □

Notice that the function θ depends only on the partial genome $\mathcal{G}$ and the reference genome $\mathcal{A}_c$, and not on the completion $\mathcal{G}_c$. Also, it is easy to see that θ is computable in linear time (in number of genes). We have the following corollary.

Corollary 3. The 1-MCP$_\ell$ is solvable in linear time.

Proof. Suppose $\mathcal{G}$ is a partial genome and $\mathcal{A}_\ell$ is a linear chromosomal reference genome. Since $\mathcal{A}_\ell$ is linear and uni-chromosomal, $|E(\mathcal{A}_\ell)| = 1$. Assume that $E(\mathcal{A}_\ell) = \{e\}$. Let $\mathcal{A}_c$ be the circular uni-chromosomal genome obtained by adding e to $A(\mathcal{A}_\ell)$. Using Algorithm 1 we obtain a solution $\mathcal{G}_c$ for 1-MCP$_c$ with $\mathcal{A}_c$ as the reference genome. Then by Theorem 3, we can transform the solution $\mathcal{G}_c$ to a linear uni-chromosomal completion $\mathcal{G}_\ell$ in linear time in the following way: If there exists an edge $f \in A(\mathcal{G}_c)\backslash A(\mathcal{G})$ such that f and e are in the same cycle of the breakpoint graph $B(\mathcal{G}_c, \mathcal{A}_c)$, i.e. $\theta(e) = 1$, remove f from $A(\mathcal{G}_c)$. Otherwise $\theta(e) = 2$ and we remove an arbitrary edge from $A(\mathcal{G}_c)$ to make a linear uni-chromosomal genome. Therefore, we obtain a solution to 1-MCP$_\ell$ by viewing $\mathcal{G}$ as a partial genome for a 1-MCP$_c$, solving the problem, and converting the solution $\mathcal{G}_c$ of 1-MCP$_c$ into a solution $\mathcal{G}_\ell$ for 1-MCP$_\ell$. Since all of these steps are done in linear time (in number of genes), the proof is complete. □

(3 ≤ *k*)-MCP

In the unrestricted case of the k-MCP, the completion of a partial genome is always possible as we can add adjacencies and telomeres arbitrarily to the partial genome, since there is no restriction on the number and type of chromosomes in the resulting genome. The hardness of showing the existence of a k-completion derives from the fact that finding a k-completion for the partial multi-genome results in a proper edge coloring for the genome graph of the partial multi-genome.

Let $G = (V, E)$ be a graph. We define the *edge-chromatic number* of G, denoted $\chi'(G)$, to be the minimum number of colors required to obtain an edge-coloring of G. For each edge-coloring of G a *color class* is a set of all edges with a specific color. A color class defines a matching in the graph since no two edges of the same color share a vertex.

The following proposition shows the relation between the edge-coloring of a genome graph and the edge color classes of the corresponding breakpoint graph.

Proposition 1. If $\mathcal{M}$ is a multi-genome of k genomes then $\chi'(\mathrm{gr}(\mathcal{M})) \leq k$.

Proof. Suppose $\mathcal{M}$ is a mixture of k genomes $\mathcal{G}_1, \dots \mathcal{G}_k$. Then the breakpoint graph $B = B(\mathcal{G}_1, \dots, \mathcal{G}_k)$ can be partitioned into the sets $A(\mathcal{G}_i)$ of adjacencies, and each $A(\mathcal{G}_i)$ can be considered as color class. So the edges of B can be colored with k colors. Since B and $\mathrm{gr}(\mathcal{M})$ are isomorphic, we have $\chi'(\mathrm{gr}(\mathcal{M})) \leq k$. □

Using the same argument as in Proposition 1 we have:

Lemma 1. If $\mathcal{M}$ is a partial multi-genome of k partial genomes then $\chi'(\mathrm{gr}(\mathcal{M})) \leq k$.

Now, in the following theorem we show a relation between the edge-coloring of a genome graph and the k-completion of the corresponding partial multi-genomes.

Theorem 4. Let $\mathcal{M}$ be a partial multi-genome. Then $\mathcal{M}$ has an unrestricted k-completion if and only if $\chi'(\mathrm{gr}(\mathcal{M})) \leq k$, for any positive integer k.

Proof. (⇒) If $\mathcal{M}$ has a k completion, then it can be considered as a partial multi-genome of k genomes. Then by Lemma 1 we have $\chi'(\mathrm{gr}(\mathcal{M})) \leq k$.

(⇐) Now assume that $\chi'(\mathrm{gr}(\mathcal{M})) \leq k$. Hence, we can color the edges of $\mathrm{gr}(\mathcal{M})$ with k colors. If $C_1, \dots, C_k$ are the color classes of G, we have $\sqcup_{i=1}^{k} C_i = E(G)$. Each C_i is a matching in the graph $\mathrm{gr}(\mathcal{M})$, and is a set of adjacencies among the gene extremities. So we can define a partial genome $\mathcal{G}'_i$ by adjacencies $A(\mathcal{G}'_i) = C_i$. The color classes partition the edges of $\mathrm{gr}(\mathcal{M})$ into k matchings, and we have $\mathcal{M} = \sqcup_{i=1}^{k} \mathcal{G}'_i$. Since there is no restriction on the completions, taking any completion $\mathcal{G}_i$ for each $\mathcal{G}'_i$ results in a a k-completion $\mathcal{M}^k = \{\mathcal{G}_1, \dots \mathcal{G}_k\}$ for $\mathcal{M}$; because $\mathcal{M} = \sqcup_{i=1}^{k} \mathcal{G}'_i \subseteq \sqcup_{i=1}^{k} \mathcal{G}_i$. □

Now, by Theorem 4 and using the following two classic theorems, we show that deciding whether there exists a valid solution to a ($k \geq 3$)-MCP is NP-complete. For a graph G let $\Delta(G)$ be the maximum degree of G.

Theorem 5 (Vizing [14]). If G is a simple graph, $\chi'(G) = \Delta(G)$ or $\Delta(G) + 1$.

Theorem 6 (Holyler [15]). For a graph G, deciding whether $\chi'(G) = \Delta(G)$ or $\Delta(G) + 1$ is NP-complete, if $\Delta(G) \geq 3$.

Corollary 4. If $k \geq 3$, deciding whether there exists a valid solution to the unrestricted k-MCP is NP-complete.

Proof. In order to prove this corollary we reduce the edge-coloring problem to k-MCP. Suppose $G = (V, E)$ is a simple graph and $k = \Delta(G) \geq 3$. If $|V|$ is not even, add an isolated vertex so that the number of vertices in G is $2n$ for some positive integer n. Consider these $2n$ vertices as gene extremities of a set of n genes. Now, G defines a partial multi-genome $\mathcal{M}$ on these n genes, since the k-MCP is unrestricted and *any* graph can be considered as a partial multi-genome with no restriction on the chromosomal structure of its partial genomes. If there is a polynomial algorithm for k-MCP, we can input to this algorithm $\mathcal{M}$ as the partial multi-genome, along with an arbitrary distance function φ and a healthy reference $\mathcal{A}$. First, suppose the algorithm gives a valid output. Since the algorithm is polynomial, we can find a k-completion for $\mathcal{M}$ in polynomial time, and by Theorem 4, we can find an edge coloring of G with k colors in polynomial time.This implies that the $\chi'(G) \leq k$. Now if the algorithm does not give a valid output, by Theorem 4 we have $\chi'(G) > k$. This implies that the k-MCP is NP-complete, since the genome graph of a partial multi-genome is always a multigraph and the class of simple graphs is a subset of the class of multigraphs. □

Note that in Corollary 4 we only considered the unrestricted version of k-MCP. This allows us to assume that for each (multi-) graph G there exists a partial multi-genome $\mathcal{M}$ such that G and $\mathrm{gr}(\mathcal{M})$ are isomorphic.Thus, if $\bar{M} = \{\mathrm{gr}(\mathcal{M}') \mid$ for all partial multi-genomes $\mathcal{M}'\}$ and if $\bar{G}$ is the set of all multi-graphs, then $\bar{M} = \bar{G}$. However, one can restrict the k-MCP by taking a set of chromosomal conditions. Consequently we may have $\bar{M} \subsetneq \bar{G}$ such that the new restricted k-MCP is polynomially tractable for all partial multi-genomes (whose genome graph is in $\bar{M}$).

Corollary 5. If $k \geq 3$, then the unrestricted k-MCP is NP-complete.

Proof. Since in solving a k-MCP we need to find a k-completion for its partial multi-genome, by Corollary 4 the proof is complete. □

2-MCP

In this section, we prove that the unrestricted 2-MCP, and the restricted 2-MCP where all chromosomes are circular (i.e., $\mathcal{C} = \{\text{circular}\}$), are NP-complete for DCJ distance. The NP-completeness of the unrestricted 2-MCP is done by a reduction from *MAX 3-AND problem*. The MAX 3-AND is a satisfiability problem, where given a set of conjunctions, each with 3 literals, the goal is to determine an assignment of Boolean value to each variable that maximizes the number of satisfied conjunctions. Note that in 2-MCP there are only two possible topologies for the mixture tree: the *branch-tree* and *path-tree* (Figure 2-b, c).

Theorem 7. The unrestricted 2-MCP with DCJ distance is NP-complete.

In order to provide the proof of this theorem, we need the following lemmas.

Lemma 2. Suppose $\mathcal{M}$ is a partial multi-genome whose genome graph, $\mathrm{gr}(\mathcal{M})$, consists of m cycles $C_1, \dots, C_m$ with even lengths, and $\mathcal{A}$ is a reference genome which consists of ℓ edges (i.e., it has ℓ adjacencies). Assume that there are ℓ' cycles among the cycles in $\mathrm{gr}(\mathcal{M})$ such that no edge in A is connected to any of their vertices. If ℓ' >2ℓ then in every solution to the 2-MCP, the optimal mixture tree is a path-tree.

Proof. Note that in 2-MCP there are only two possible topologies for the mixture tree: the *branch-tree* and *path-tree* (Figure 2-b, c). Since the degree of each vertex in $\mathrm{gr}(\mathcal{M})$ is two, if we partition the edges of $\mathrm{gr}(\mathcal{M})$ into two perfect matchings $\mathcal{G}_1'$ and $\mathcal{G}_2'$. Therefore, for any 2-completion $\mathcal{M}^2 = \{\mathcal{G}_1, \mathcal{G}_2\}$ we have $\mathcal{G}_1' = \mathcal{G}_1$ and $\mathcal{G}_2' = \mathcal{G}_2$, since G_1 and G_2 are maximal (and circular) and we cannot add any edge to them. Also, for each $\mathcal{G}_i (i = 1, 2)$ we have $\mathcal{G}_i = \cup_{j=1}^{m} M_{ij}$, where M_{ij} is a perfect matching on vertices of C_j. Obviously, $c(\mathcal{G}_1, \mathcal{G}_2) = m$. We have $c(\mathcal{A}, \mathcal{G}_i) \leq \ell$ for i = 1, 2, since $\mathcal{A}$ has ℓ edges and each of them can be in at most one cycle in $B(\mathcal{G}_i, \mathcal{A})$. Therefore,

$$\begin{aligned} c(\mathcal{A}, \mathcal{G}_1) + c(\mathcal{G}_1, \mathcal{G}_2) \geq c(\mathcal{G}_1, \mathcal{G}_2) = m \geq \ell' > 2\ell \\ \geq c(\mathcal{A}, \mathcal{G}_1) + c(\mathcal{A}, \mathcal{G}_2), \end{aligned}$$

which shows that the d_{DCJ}-value of a path tree is smaller than the d_{DCJ} -value of a branch tree, and completes the proof. □

Lemma 3. Any MAX 3-SAT instance is reducible to a MAX 3-AND instance. Moreover, MAX 3-AND is NP-complete.

Proof. Let Δ = ℓ_1 V ℓ_2 V ℓ_3 be a clause (disjunction) of three literals. Define

$$L(\Delta) = \{(t_1 \wedge t_2 \wedge t_3) | 1 \leq i \leq 3, t_i \in \{\ell_i, \bar{\ell}_i\}, (t_1, t_2, t_3) \neq (\bar{\ell}_1, \bar{\ell}_2, \bar{\ell}_3)\}.$$

By using basic Boolean rules we have Δ ⇔ $\mathrm{V}_{S \in \ell(\Delta)} S$.

Now, suppose $\mathcal{I}$ is a MAX 3-SAT instance which has m clauses $\Delta_1, . . ., \Delta_{m.}$ Let $\mathcal{I}'$ be an instance of MAX 3-AND which consists of all the conjunctions in $\cup_{j=1}^{mL}(\Delta_j)$. Since for every assignment to the variables at most one conjunction in $L(\Delta_j)$, $1 \leq j \leq m$, is satisfied and this happens if and only if Δ_j is satisfied, then every optimal assignment to the variables in $\mathcal{I}$ will be also an optimal assignment to the variables in $\mathcal{I}$. Therefore, MAX 3-SAT is reducible to MAX 3-AND, which implies that MAX 3-AND is NP-complete, as MAX 3-SAT is NP-complete [16]. □

Now, consider an instance $\mathcal{I}$ of the MAX 3-AND problem. We show how to represent $\mathcal{I}$ by a genome graph and a reference genome, to make a reduction from MAX 3-AND to 2-MCP. Suppose we represent a variable x with a cycle C of even length, which we will call a *variable-cycle* (see Figure 6-a). This cycle has exactly two perfect matchings. We label one of these the *true* matching, $T(x)$, and the other one the *false* matching, $F(x)$ (see Figure 6-b, c). We represent an assignment to a variable by choosing one of the matchings $T(x)$ and $F(x)$ and **remove** the edges in the other matching (see Figure 7).

Let $\ell(x_1)$, $\ell(x_2)$, $\ell(x_3)$ be three literals of variables x_1, x_2, x_3, and $\Delta = (\ell(x_1) \wedge \ell(x_2) \wedge \ell(x_3))$ be a conjunction in $\mathcal{I}$. A *conjunction-cycle* of Δ is a cycle which is obtained as follows:

1. For each $i \in \{1, 2, 3\}$ consider an edge in $T(x_i)$ if $\ell(x_i) = x_i$. If $\ell(x_i) = \bar{x}_i$ take an edge in $F(x_i)$.
2. Add three new edges, called *conjunction-edges*, to the three edges we chose in the previous step, and build a cycle of length 6. This cycle is a conjunction-cycle of Δ.

It is easy to see that an assignment α to x_i's satisfy the conjunction Δ if and only if the corresponding matching assignment to α keeps all the edges in the conjunction-cycle of Δ. We call a representation of a MAX 3-AND instance $\mathcal{I}$ with cycles and conjunction-cycles *a graphical representation of* $\mathcal{I}$.

If the literals of a variable appear in at most t conjunctions, and the variable-cycles have length at least $4t$, then by choosing the edges of conjunction-cycles properly, we have a graphical representation of a MAX 3-AND instance, where no edge in a variable-cycle is incident with two conjunction edges from different conjunction-cycles. This implies the following lemma:

Lemma 4. For each MAX 3-AND instance $\mathcal{I}$ there exists a graphical representation $\mathcal{I}_g$ such that any assignments to the variables in $\mathcal{I}$ which maximizes the number of satisfied conjunctions, induces a matching assignment that maximizes the number of conjunction-cycles, and vice versa.

Combining Lemmas 2-4 gives the proof of Theorem 7.

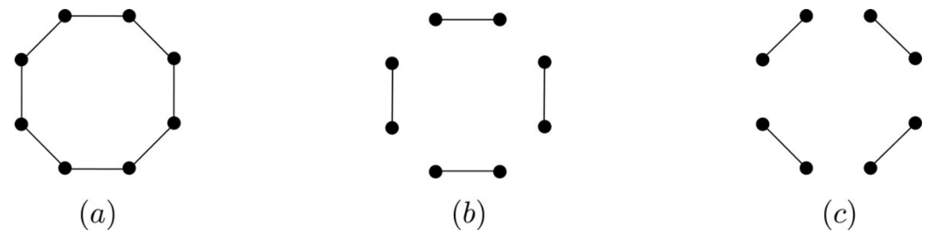

Figure 6 Representing variables with cycles. (a) A variable represented by a cycle, (b) a `true` matching, and (b) a `false` matching.

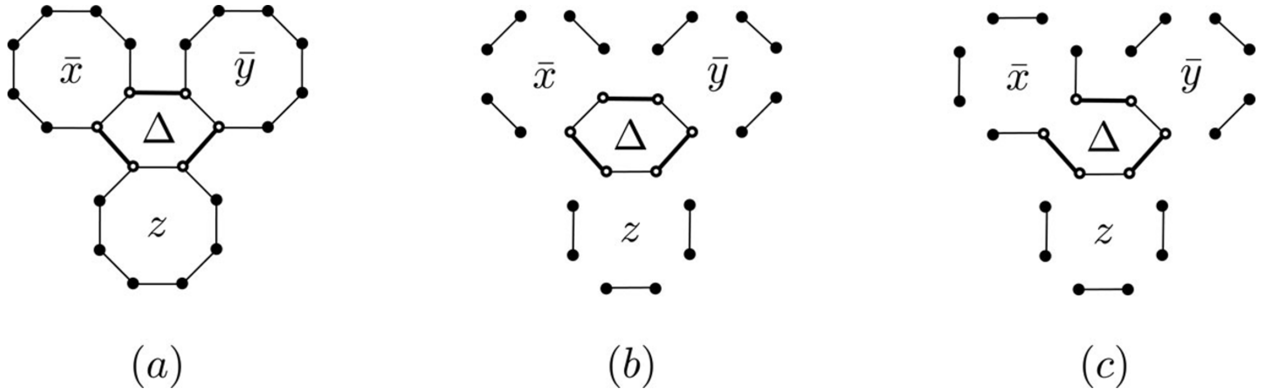

Figure 7 Representing conjunctions with cycles. (a) Three cycles representing the literals $\bar{x}$, $\bar{y}$, and z, and the conjunction edges (bold) for a conjunction $\Delta = \bar{x} \wedge \bar{y} \wedge z$. (b) For $x = y =$ `false` and $z =$ `true` we obtain the conjunction-cycle Δ of length 6. (c) Any other assignment (e.g., $x =$ `true`) destroys the conjunction cycle.

Proof of Theorem 7. Since the MAX 3-AND is NP-complete by Lemma 3, it suffices to reduce the MAX 3-AND problem to the 2-MCP. Suppose $\mathcal{I}$ is a MAX 3-AND instance. Assume $\mathcal{I}$ has m conjunctions. We can add $3m + 1$ new conjunctions $\delta_1, \ldots, \delta_{3m+1}$ where each δ_i consists of a new single variable $x_{\delta i}$; obviously in any optimal assignment the value of all the $x_{\delta i}$'s should be `true`. Now by Lemma 4, there is a graphical representation $\mathcal{I}_g$ such that finding an optimal assignment in $\mathcal{I}$ is equivalent to finding a matching for each variable-cycle such that the number of preserved conjunction-cycles are maximized. Note that there are $3m$ conjunction-edges and $3m + 1$ variable-cycles which are not connected to any conjunction-edge. Now, consider all the vertices in $\mathcal{I}_g$ as gene extremities, and all the edges in the variable-cycles as the adjacencies of a partial multi-genome G. Also, consider all the conjunction-edges as the adjacencies of a reference healthy genome $\mathcal{A}$. In the 2-MCP problem with partial multi-genome G and reference healthy genome $\mathcal{A}$, the optimal tree is forced to be a path-tree by Lemma 2 (Figure 2). Therefore, in the optimal solution of this 2-MCP, $\mathcal{G}_1$ should be a genome such that the number of cycles in the breakpoint graph $B(\mathcal{G}_1, \mathcal{A})$ is maximized, i.e., the number of conjunction-cycles are maximized. Since $\mathcal{G}_1$ is a union of perfect matchings of the variable-cycles (see the proof of Lemma 2) it induces an assignment for the variables which maximizes the number satisfied conjunctions, and this completes the proof. □

We end this section by considering the restricted version of k-MCP, where the chromosomal condition set is {circular}, i.e. all genomes have all circular chromosomes. We denote this restricted version by k-MCP$_c$, and the unrestricted version of k-MCP by k-MCP$_\varnothing$. If opt(k-MCP$_c$) and opt(k-MCP$_\varnothing$) are the d_{DCJ}-value of a solution to k-MCP$_c$ and k-MCP$_\varnothing$, respectively, then:

Theorem 8. For the k-MCP$_c$ and k-MCP$_\varnothing$ versions of k-MCP with DCJ distance we have

$$\mathrm{opt}(k - \mathrm{MCP}_c) = \mathrm{opt}\,(k - \mathrm{MCP}_\emptyset)\,.$$

Proof. First note that each solution to k-MCP$_c$ is also a solution of k-MCP$_\varnothing$, since there is no restriction in k-MCP. Hence, opt(k-MCP$_c$) ≥ opt(k-MCP$_\varnothing$). Second, for each solution to k-MCP$_\varnothing$ if the resulting genomes are not circular we can add new edges to the genomes and make them circular. By adding the new edges the number of cycles in the breakpoint graph does not decrease which implies that the d_{DCJ}-value of the newly obtained genomes is not larger than opt(k-MCP$_\varnothing$). Therefore, these circular genomes form a solution of k-MCP$_\varnothing$. So opt(k-MCP$_c$) ≤ opt(k-MCP$_\varnothing$) completing the proof. □

Combining this theorem and Theorem 7 we have

Corollary 6. If $k \geq 2$, then k-MCP$_c$ with DCJ distance is NP-complete.

Discussion and conclusion

In this paper we introduced the k-Minimum Completion Problem (k-MCP) motivated by the type of data produced in current cancer genome sequencing studies. We showed the following results. (1) A linear time algorithm for the unrestricted 1-MCP; (2) a linear time algorithm for the restricted versions 1-MCP where the genomes are circular or linear; i.e. the chromosomal condition set $\mathcal{C}$ is {circular, uni-chromosomal} or $\mathcal{C}$ is {linear, uni-chromosomal}; (3) the unrestricted k-MCP is NP-complete for any distance when $k \geq 3$; and (4) the 2-MCP with DCJ distance is NP-complete in the unrestricted version and with the condition that all chromosomes are circular, i.e. $\mathcal{C}$ = {circular}. These results lay the foundation for future algorithmic studies of the k-MCP and the application of these algorithms to real cancer sequencing data.

There are numerous further directions to pursue. As noted in the introduction, the model described in this paper does not consider all the complexities of cancer genome sequencing: most importantly copy number aberrations (duplications and deletions) and errors in the measured adjacencies are important features of cancer genome sequencing and should be addressed.

To handle errors, one might consider weighted versions of the *k*-MCP where adjacencies have a weight corresponding to the confidence in the measurement. Regarding the current model, further work is needed on different chromosomal conditions, genomic distances, or other constraints on the relationships between the genomes in the mixture. For example, the case of linear chromosomes demands further attention, as human chromosomes are linear, although circular chromosomes have been observed in cancer [17]. Similarly, one may impose an upper bound on the number of chromosomes. One may also place restrictions on the structure of the mixture tree.

Another direction is to derive approximation algorithms. In the *k*-MCP we aim to minimize distance over all possible *k*-completion and mixture trees simultaneously. However, by separating the completion and distance optimization steps, one may employ techniques that have developed for other problems. For example, one may try to first complete the partial multi-genomes using some clustering techniques, as have been employed in metagenomic studies [18]. With complete genomes, one could then try to find optimal mixture trees rooted at the reference genome. Depending on the allowed structure of the mixture tree, techniques from genome rearrangement phylogeny problems may be employed. For example, in the case of 2-MCP, if the complete genomes are the leaves of the mixture tree, then the problem becomes the *median problem* (with the reference genome genome as the third genome) [5,13]. Alternatively, if the genomes are the vertices of the mixture tree, then the tree construction problem becomes the problem of finding a minimum spanning tree, which is in generally easier. In between these extremes, where some of the genomes in the mixture are the leaves and some are intermediate nodes (ancestors), the problem becomes a Steiner tree problem. In the cancer application, any of these cases might provide useful approximations, as the process of clonal evolution of cancer [1] might mean that cells at intermediate stages of cancer progression remain present in the tumor.

Acknowledgements

We thank the anonymous referees for helpful comments on an earlier version of this manuscript. This work was supported by a CAREER Award from the National Science Foundation (#1053753). In addition, BJR is supported by a Career Award from the Scientific Interface from the Burroughs Wellcome Fund and an Alfred P. Sloan Research Fellowship.

This article has been published as part of *BMC Bioinformatics* Volume 13 Supplement 19, 2012: Proceedings of the Tenth Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/13/S19.

Authors' contributions

All authors contributed equally to this work.

Competing interests

The authors declare that they have no competing interests.

Published: 19 December 2012

References

1. Nowell PC: **The clonal evolution of tumor cell populations.** *Science* 1976, **194(4260)**:23-28.
2. Raphael BJ, Volik S, Collins C, Pevzner PA: **Reconstructing tumor genome architectures.** *Bioinformatics* 2003, **19(Suppl 2)**:i162-171.
3. Meyerson M, Gabriel S, Getz G: **Advances in understanding cancer genomes through second-generation sequencing.** *Nat Rev Genet* 2010, **11(10)**:685-696.
4. Yancopoulos S, Attie O, Friedberg R: **Efficient sorting of genomic permutations by translocation, in-version and block interchange.** *Bioinformatics* 2005, **21(16)**:3340-3346.
5. Tannier E, Zheng C, Sankoff D: **Multichromosomal median and halving problems under different genomic distances.** *BMC Bioinformatics* 2009, **10.**
6. Hannenhalli S, Pevzner PA: **Transforming Men into Mice (Polynomial Algorithm for Genomic Distance Problem).** *FOCS* IEEE Computer Society; 1995, 581-592.
7. Hannenhalli S, Pevzner PA: **Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals.** *J ACM* 1999, **46**:1-27.
8. Bergeron A, Mixtacki J, Stoye J: **A new linear time algorithm to compute the genomic distance via the double cut and join distance.** *Theor Comput Sci* 2009, **410(51)**:5300-5316.
9. El-Mabrouk N, Sankoff D: **The Reconstruction of Doubled Genomes.** *SIAM J Comput* 2003, **32(3)**:754-792.
10. Warren R, Sankoff D: **Genome Aliquoting Revisited.** In *RECOMB-CG, Volume 6398 of Lecture Notes in Computer Science.* Springer;Tannier E 2010:1-12.
11. Zheng C, Lenert A, Sankoff D: **Reversal distance for partially ordered genomes.** *ISMB (Supplement of Bioinformatics)* 2005, 502-508.
12. Gaul É, Blanchette M: **Ordering Partially Assembled Genomes Using Gene Arrangements.** In *Comparative Genomics, Volume 4205 of Lecture Notes in Computer Science.* Springer;Bourque G, El-Mabrouk N 2006:113-128.
13. Xu AW: **A Fast and Exact Algorithm for the Median of Three Problem-A Graph Decomposition Approach.** In *RECOMB-CG, Volume 5267 of Lecture Notes in Computer Science.* Springer;Nelson CE, Vialette S 2008:184-197.
14. Vizing VG: **On an estimate of the chromatic class of a *p*-graph. (Russian).** *Diskret Analiz* 1964, **3**:25-30.
15. Holyer I: **The NP-Completeness of Edge-Coloring.** *SIAM J Comput* 1981, **10(4)**:718-720.
16. Cook SA: **The Complexity of Theorem-Proving Procedures.** In *STOC.* ACM; Harrison MA, Banerji RB, Ullman JD 1971:151-158.
17. Raphael BJ, Pevzner PA: **Reconstructing tumor amplisomes.** *ISMB/ECCB (Supplement of Bioinformat-ics)* 2004, 265-273.
18. Wu YW, Ye Y: **A Novel Abundance-Based Algorithm for Binning Metagenomic Sequences Using ℓ-Tuples.** In *RECOMB, Volume 6044 of Lecture Notes in Computer Science.* Springer;Berger B 2010:535-549.

doi:10.1186/1471-2105-13-S19-S9
Cite this article as: Mahmoody *et al.*: Reconstructing genome mixtures from partial adjacencies. *BMC Bioinformatics* 2012 **13**(Suppl 19):S9.