The Minimum Letter Flip Problem for Haplotyping a Single Individual

John Louie
Carroll College, Helena, MT

Follow this and additional works at: https://scholars.carroll.edu/mathengcompsci_theses

Part of the Applied Mathematics Commons, Genetics Commons, and the Genomics Commons

Recommended Citation
https://scholars.carroll.edu/mathengcompsci_theses/69

This Thesis is brought to you for free and open access by the Mathematics, Engineering and Computer Science at Carroll Scholars. It has been accepted for inclusion in Mathematics, Engineering and Computer Science Undergraduate Theses by an authorized administrator of Carroll Scholars. For more information, please contact tkratz@carroll.edu.
The Minimum Letter Flip Problem for Haplotyping a Single Individual

John Louie

An Honors Thesis Submitted to the Department of Mathematics in Partial Fulfillment of the Requirements for Graduation with Honors

Carroll College
Helena, MT

April 6, 2006
This thesis for honors recognition has been approved for the Department of MATHEMATICS.

Director: [Signature]
Date: 4/6/06

Reader: [Signature]
Date: 5 April 2006

Reader: [Signature]
Date: 4/4/06
# Contents

1 Abstract 3

2 Acknowledgements 4

3 Introduction 5
   3.1 Haplotyping a Single Individual 5

4 Notation and Problem Statements 8
   4.1 P-median Problem 8
   4.2 SNP Matrix 8
   4.3 d-Distance Measure 9
   4.4 l-Distance Measure 9
   4.5 Median Sets 9
   4.6 Problem Formulations 11

5 MLF' 12
   5.1 MLF' Algorithm 12
   5.2 Complications 14

6 DMLF' and DMLF 17
   6.1 DMLF' Algorithm 18
   6.2 Results 19

7 Relating DMLF to CGMLF 21

8 Relating Each Problem Formulation 25

9 Conclusion and Future Work 26
   9.1 Future Work 26
1 Abstract

When haplotyping a single individual, DNA is replicated, broken into smaller fragments, and then sequenced by a machine. The minimum letter flip problem is one approach to correcting errors that arise in shotgun sequencing. We refer to the minimum letter flip problem as stated in other texts as CGMLF. CGMLF changes a minimal number of single nucleotide polymorphisms (SNPs) to create a feasible SNP matrix. Since finding partitions that have this property is especially difficult, we relate CGMLF to several 2-median problem formulations. Our major result is that the CGMLF is equivalent to a non-polynomial 2-median problem formulation. We develop algorithms used to solve the 2-median problems and discuss their complexity. In conclusion, we develop inequality relations for our problem formulations based on minimum number of flips and prove that CGMLF can be bounded from above in polynomial time.
2 Acknowledgements

This research has been done under the Research Experience for Undergraduates (REU) program at Trinity University supported by the National Science Foundation. I would like to thank Dr. Allen Holder for his help as my advisor at Trinity University and Lena Sherbakov for her contributions as a group member on this project.

I would like to thank Dr. Holly Zullo for her work while advising this thesis and her efforts in encouraging me to participate in the REU. I would also like to thank Mr. Philip Rose and Dr. Jeffrey Morris for contributing as readers to this thesis.

Finally, I would like to thank my parents, Robert and Christine, for their continued support of my education. Their love and encouragement in all aspects of my life have meant the world to me.
3 Introduction

With numerous opportunities arising in the wake of the human genome project, the ability to accurately reconstruct an individual DNA sequence is becoming increasingly important. The benefits of this extend beyond forensic applications into drug design and many other medical and related fields. In this paper, we investigate ways to correct errors that naturally occur during DNA sequencing.

Deoxyribonucleic acid (DNA) is a molecule found in the cell nucleus that is responsible for many of our inherited traits. DNA consists of approximately 3 billion smaller molecules called nucleotides. When we sequence a strand of DNA, we are determining the exact order of these nucleotides. This information becomes important when we analyze the genes of an individual. A gene is a fixed region of DNA generally considered to contain a hereditary trait, such as flower blossom color or human blood type. The variations of the gene are called alleles. Using blood type as an example, there is an allele for type A and type B, and when an individual has both alleles they combine to yield the AB blood type.

When a DNA strand is sequenced, a string of A’s, T’s, G’s, and C’s is generated that represent the nucleotides of the DNA molecule. The letter representations from the sequencer coincide with the nucleotides Adenine, Thymine, Guanine, and Cytosine that form the DNA molecule. A sequencer is not capable of handling an entire strand of DNA and, therefore, a process known as shotgun sequencing is used. The long DNA strand is replicated several times and these copies are divided into random fragments of about 1,000 to 30,000 individual nucleotides [3]. The fragments are then sequenced in the machine and then aligned to reconstruct the original genomic sequence.

Humans are diploid organisms with pairs of chromosomes: one paternal and one maternal. Aligning the DNA fragments from the sequencer is difficult since most of the fragments from both donations are nearly identical. Geneticists consider a Single Nucleotide Polymorphism, or SNP, to make distinctions between the two parental strands. SNPs (pronounced “snips”) are single nucleotide differences where we observe a statistically increased level of variability [3]. A SNP can be either homozygous (same on both chromosomes) or heterozygous (different nucleotides) in a diploid organism.

3.1 Haplotyping a Single Individual

A haplotype is a set of polymorphisms in a region that tend to be inherited together because of their proximity on the genome. When haplotyping an individual, we are trying to obtain a coherent pair of parental SNP haplotypes. In this manner, a haplotype is a string of SNPs and excludes the many sections of DNA that are nearly identical. This process is complicated by errors that arise as misread SNPs or skipped data, as well as from another organism inadvertently contaminating multiple fragments [3]. Several methods to correct these errors have been suggested, ranging from
the minimum fragment removal (MFR), minimum SNP removal (MSR), and the problem we investigate, the minimum letter flip (MLF) problem [3].

A standard way to organize the data is to create a SNP matrix, where each row denotes a fragment and each column denotes a SNP location. Since diploid organisms have two different strands of DNA and, therefore, only two possible alleles at each SNP, we say that a SNP is either an A or B. A SNP may also be labeled as a — in the instances where the sequencer could not call a position with enough certainty. A $m \times n$ SNP matrix is defined over the set of fragments $\{1, ..., m\}$ and the set of SNPs $\{1, ..., n\}$. An example of a SNP matrix is found in Table 1.

We use $h^i$ to denote fragment $i$ in the haplotype set $H$. Fragment $h^i$ is in conflict with $h^j$ if at any SNP location $p$, $h^i_p$ and $h^j_p$ differ, with the case of a "—" being handled differently depending on the problem. The MFR determines the minimum number of fragments to remove in order to create a resulting SNP matrix that is feasible, meaning that the fragments can be divided into two disjoint sets where the fragments within each set are conflict-free. Each set can then be used to derive the entire parent DNA strand. To illustrate the definition of feasibility, construct a conflict graph from the fragments in Table 1. See Figure 1.

Figure 1(a) shows the conflict graph of Table 1. An edge between vertices means that there is a conflict between the two fragments. To make this graph feasible, the MFR problem removes nodes from this graph until the resulting subgraph is bipartite. The partitioning of the SNP matrix in Table 1 is shown in Figure 1(b). In this case, removing fragments 1 and 4 creates a bipartite graph, with the first collection of fragments containing fragments 3 and 5 and the second collection consisting of fragments 2 and 6 (Figure 1(c)). The two parental chromosomes and their copies are represented by the two disjoint sets. The MFR approach is commonly used in situations where a contaminant may be present and entire fragments need to be removed to correct the data.

The MSR problem removes the minimum number of SNPs to create a feasible matrix. Instead

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragments</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>-</td>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>A</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>A</td>
<td>-</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 1: SNP Matrix
of removing rows of the SNP matrix, the MSR problem removes columns from the original matrix. This would be the process to use when only eliminating sequencing errors is appropriate. Using the SNP matrix from Table 1, removing columns 1 and 2, for example, would create a feasible solution where fragments 2, 5 and 6 belong in one set and 1, 3 and 4 belong to the other.

The problem we investigate is the MLF problem. Given a SNP matrix, we want to flip (change) the minimum number of SNPs to create a resulting feasible matrix. To better understand the algorithms discussed in this paper, an understanding of Big-O notation is essential. We say that a function \( f(n) \) is \( O(g(n)) \) if there exists constants \( c \) and \( k \) such that for all \( n > k \), \( |f(n)| \leq c(|g(n)|) \). For example, consider the case where \( f(n) = 2n \) and \( g(n) = n \). We would say \( f(n) \) is \( O(g(n)) \) because we can let \( c = 3 \) and \( k = 0 \), since for all \( n > 0 \), \( 2n \) is \( \leq 3(g(n)) \) or \( 3n \). Big-O can compare two polynomial functions by their highest-order term because the leading term dominates the function. In this paper, Big-O notation is used to describe the number of iterations an algorithm requires to generate a solution.

The following section details each of the different problem formulations we use to approach the MLF problem. We explain how they relate and how each provides insight into our goal of flipping the minimal number of letters to create a feasible solution. Subsequent sections discuss algorithms used to solve our problem formulations, Big-O results, and relations between different problem statements based on minimum number of flips. We conclude the paper by stating a few open problems and areas of future work.
4 Notation and Problem Statements

The minimum letter flip problem is defined to be the smallest number of flips necessary to create a resulting feasible matrix. Since a feasible matrix implies the creation of a conflict graph, we refer to this original problem statement as the CGMLF. The CG stands for conflict graph, and we say that the CG(H) is the conflict graph of the elements in a set H. We use the notation z(H, Ĥ) to be the number of total flips required to create the feasible partition set  Ĥ from the set H. We formally define the CGMLF as follows:

**Definition 4.1.** The CGMLF problem is a relabelling problem that transforms H into Ĥ such that CG(H) is bipartite and z(H, Ĥ) is as small as possible.

A relabelling of an h^i ∈ H to an ̂h^i is used when a flip has changed the representation of a fragment h^i. Since the CGMLF is a partitioning problem that is known to be NP-hard, an easier problem to approach is the 2-median problem, which chooses medians before creating partitions. We discuss the 2-median approaches to this problem next.

4.1 P-median Problem

We address the CGMLF problem by relating it to the well-known p-median problem. The p-median problem is stated as follows[1]:

Locate p facilities in a network so that the sum of the distances between every client in the network and its nearest facility is minimized.

By recognizing what is meant by a facility, network, client, and distance from the definition above, we will rewrite this p-median problem in a way that helps us understand the MLF problem. In this case, there are two facilities that represent the two optimal haplotypes from a set of fragments and we consider two definitions of the distance between clients, as discussed below.

4.2 SNP Matrix

Given our definition of a SNP and a fragment, we take m fragments with n SNP entries in {A, B, −} and construct a m × n SNP matrix. The set of fragments in the SNP matrix is,

\[ H = \{h^1, h^2, ..., h^m\}. \]

(Note: We use a superscript to denote a specific fragment and a subscript for a specific SNP location on that fragment.) We then assign each of the \( \binom{m}{2} \) pairs of fragments a distance.
4.3 \textit{d}-Distance Measure

The first formula for a distance measure is a Hamming distance defined as the number of SNP locations at which $h^i$ and $h^j$ vary without penalty paid for going from a $-$ to a letter or from a letter to a $-$. This distance is defined as,

\[ d(h^i, h^j) = \sum_{p=1}^{n} d(h^i_p, h^j_p), \text{ where } d(h^i_p, h^j_p) = \begin{cases} 1, & \text{if } h^i_p = A, h^j_p = B \\ 1, & \text{if } h^i_p = B, h^j_p = A \\ 0, & \text{otherwise.} \end{cases} \]

We will call this distance the \textit{d}-distance. As an example, the \textit{d}-distance between the fragment ABB--B and --BAA-- is 1 (from the third SNP location). Notice that $B$ and --, $A$ and --, and -- and -- are not considered conflicting SNPs.

It is of interest to note that the \textit{d}-distance would not be considered a metric space because the triangular inequality does not always hold. Consider the three fragments $h^1 = AB$, $h^2 = A-$, and $h^3 = AA$. The distance from $h^1$ to $h^2$ is 0 and the distance from $h^2$ to $h^3$ is 0 as well. From $h^1$ to $h^3$, however, the distance is 1. This results in a contradiction to the triangular inequality, that is, it is defined to be $d(x, y) + d(y, z) \geq d(x, z)$.

4.4 \textit{l}-Distance Measure

The second formulation of a distance measure is obtained by penalizing a move from $A$ or $B$ to --.

\[ l(h^i, h^j) = \sum_{p=1}^{n} l(h^i_p, h^j_p), \text{ where } l(h^i_p, h^j_p) = \begin{cases} 1, & \text{if } h^i_p = A, h^j_p = B \\ 1, & \text{if } h^i_p = B, h^j_p = A \\ 1, & \text{if } h^i_p \in \{A, B\}, h^j_p = - \\ 0, & \text{otherwise.} \end{cases} \]

We call this distance the \textit{l}-distance. As an example, $l(h^i, h^j)$ with $h^i = ABB--B$ and $h^j = --BAA--$ is 3 (from the first, third, and last locations). However, $l(h^j, h^i)$ is only 2 (from the third and fourth SNPs). Notice that there is no penalty for turning a $-$ into either an $A$ or $B$ with this distance measure. Taking an $A$ or $B$ and changing it into a $-$ is penalized because we are changing a known SNP into an unknown.

This distance measure is also not a metric space. From the preceding example, we can see that the symmetry rule fails for all cases. That is, $l(x, y)$ may not equal $l(y, x)$.

4.5 Median Sets

The distance formulas apply to fragments outside of $H$ as well. In many of the problem formulations, we need to know the distance between any possible pair of fragments. We will define

\[ S_n = \{(s^1, s^2, ..., s^n) : s^i \in \{A, B, -\} = \{A, B, -\}^n. \]
This is simply the set of all possible fragments (not necessarily in H) with n SNPs. Some important relations on \( S_n \) are defined below, where \( s^i \) is the \( i^{th} \) fragment and \( s^i_k \) is the \( k^{th} \) position in the \( i^{th} \) fragment:

\[
\begin{align*}
& s^i = s^j \text{ iff } s^i_k = s^j_k, \forall k. \\
& s^i \sim s^j \text{ iff } s^i_k = s^j_k \text{ or } s^i_k = - \text{ at any } k.
\end{align*}
\]

An example of the relation \( \sim \) is the following:

Consider \( h^i = ABBA-, h^j = A-B--\), and \( h^k = AB-AA\).

We say that \( h^j \sim h^k \), but \( h^k \) is not \( \sim h^i \) because the last SNP is defined in \( h^k \) but not \( h^i \).

Another important set to define is,

\[
S'_n = \{(s_1, s_2, \ldots, s_n) : s_i \in \{A, B\}\} = \{A, B\}^n.
\]

This is the set of all fragments with n SNPs, where each SNP is either an A or B. Every fragment in \( S'_n \) is a complete sequence of A's and B's without any '-'s.

After assigning a distance between all fragments, we look for a pair of fragments, \((\gamma^1, \gamma^2)\), that represent one solution to the 2-median problem. Depending on the problem formulation, \((\gamma^1, \gamma^2)\) may or may not be elements in \( H \). Alternatively, we think of \((\gamma^1, \gamma^2)\) as parent haplotypes, which may not necessarily be unique due to equivalent 2-median solutions. Note that this notation does not imply that \( \gamma^1 \) is the first fragment in \( H \). After having located \((\gamma^1, \gamma^2)\), each \( h^i \) is then assigned to the nearest median. The nearest median from a fragment \( h^i \) is the median \( \gamma^i \in \{\gamma^1, \gamma^2\} \) of minimal distance away. In addition, \( \Gamma^r \) is the set of all fragments whose nearest median is \( \gamma^i \). With these new definitions, we now reformulate the 2-median problem as follows:

**Definition 4.2. General 2-Median Problem on a Complete Graph:**

Let \( K_H \) be the complete graph on the nodes of \( H \), and \( U \subseteq S_n \). Let \( E(H, U) = \{(v^i, v^j) : v^i \in H, v^j \in U\} \), where each edge is assigned a weight \( w_{ij} \). The general 2-median problem is to find \( M = (\gamma^1, \gamma^2) \) such that:

1) \( \Gamma^1 \cup \Gamma^2 = H \).
2) \( \Gamma^1 \cap \Gamma^2 = \emptyset \).
3) \( \sum w_{ij} \forall i \text{ and } j \) is minimized.

The following four problem formulations are all 2-median problems. To better understand the acronyms, when a ' follows a title, we are referring to a 2-median problem where only fragments within \( H \), the original set of fragments, are considered as possible medians. A problem without the ' allows a median to be from the set \( S_n \). The MLF and MLF' use the d-distance and the DMLF and DMLF' use the l-distance.
### 4.6 Problem Formulations

We first look at the $MLF'$ problem formulation.

**Definition 4.3.** $MLF'$ is the 2-median problem on $K_H$ with $w_{ij} = d(v^i, v^j)$.

The $MLF'$ problem locates the two medians from the given set of haplotypes $H$ using the $d$-distance. Fundamentally, the $MLF'$ selects two fragments that combine to have the least number of conflicts with the remaining fragments. The problem is on $K_H$, which means the medians are located from the set of fragments in $H$.

Looking at Table 2, the $MLF'$ would return fragments 1 and 2 as the medians of this SNP matrix. These are the only two fragments within the set that are zero distance away from their assigned fragments. In this example, $\gamma^1 = h^1$ where $\Gamma^{\gamma^1} = \{h^1, h^3, h^6, h^8\}$ and $\gamma^2 = h^2$ where $\Gamma^{\gamma^2} = \{h^2, h^3, h^5, h^7\}$. Note that the assignments of $h^7$ and $h^8$ were arbitrary.

**Definition 4.4.** $MLF$ is the 2-median problem on $(S_n, E(H, S_n))$ with $w_{ij} = d(v^i, v^j)$.

This problem is similar to the $MLF'$ problem except that it allows a median to be outside of the given set $H$. While still using the $d$-distance, the $MLF$ will find two medians that minimize the distance from all the fragments in $H$ to the medians found in $S_n$. The problem is on $(S_n, E(H, S_n))$, which means the medians are located over the edge set from all fragments in $H$ to the fragments in $S_n$. Recall that $S_n$ is all possible fragments of $n$ SNPs, so $H \subseteq S_n$.

Using Table 2 as an example, the $MLF$ could return fragments 1 and 2 as medians, but it is easy to see that the $MLF$ could also return a fragment of all $-$'s and any fragment from the set $H$ and it would still have a total distance of zero. Due to this property, the $MLF$ will trivially return the all $-$ fragment with a flip count of zero. Clearly this isn’t a very useful problem formulation, but we’ve included it for completeness.

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: SNP Matrix
Definition 4.5. DMLF' is the 2-median problem on \( (K_H = (H, E(H, H))) \) with \( w_{ij} = l(v^i, v^j) \).

The DMLF' is similar to the MLF', but the \( l \)-distance is used to determine the edge weight rather than the \( d \)-distance. The two medians are located within the given haplotype set \( H \).

In Table 2, the DMLF' would return fragments 3 and 4 as the medians where \( \sum w_{ij} = 11 \). This means that 11 flips are required to make \( K_H \) bipartite. \( \gamma^1 = h^3 \) and \( \Gamma^1 = \{h^1, h^3, h^8\} \) and \( \gamma^2 = h^4 \) where \( \Gamma^2 = \{h^2, h^4, h^5, h^6, h^7\} \). While this flip count may seem high, most of the flips are acquired when a known SNP is relabelled as a – in order to be conflict-free with its median. The \( l \)-distance, as a reminder, penalizes changing a known SNP into an unknown, and here we are finding the distance from a fragment to its median.

Definition 4.6. DMLF is the 2-median problem on \( (S_n, E(H, S_n)) \) with \( w_{ij} = l(v^i, v^j) \).

The DMLF is locating the two medians from the set \( S_n \) and uses the \( l \)-distance. This allows DMLF to search for the two parent haplotypes from the entire set of possible fragments, and uses the more biologically relevant distance formula as well.

Looking at Table 2, the DMLF would return the fragments \{ABAABAA, BABBABB\} as the medians with three required flips, where \( \Gamma^1 = \{h^1, h^3, h^6, h^8\} \) and \( \Gamma^2 = \{h^2, h^4, h^5, h^7\} \). These flips occur when we relabel the fifth SNP location on the 4\(^{th}\) fragment, the sixth SNP on the 6\(^{th}\) fragment, and the seventh SNP on the 7\(^{th}\) fragment.

In what follows, we take a closer look at these problem formulations and the algorithms used to solve them. We discuss the strengths and weaknesses of each 2-median problem and relate them to our true problem formulation, the CGMLF that is simply creating a bipartite solution of the given haplotype set.

5 MLF'

Our first investigation into the minimum letter flip problem is with the MLF'. As a reminder, the MLF' locates medians from the given haplotype set \( H \) and uses the \( d \)-distance to compare fragments.

5.1 MLF' Algorithm

The algorithm presented here determines both the minimal number of flips and a set of medians \( \{\gamma^1, \gamma^2\} \) that solves the MLF' problem.
ALGORITHM 1

Step 0: Set min = \(\infty\).

Step 1: Arbitrarily number the fragments 1 through \(m\).

Step 2:  
- for \((a = 1, m)\)
  - for \((b = a + 1, m)\)
    * flip count = 0
      - for \((c = 1, m)\)
        - if \(d(a, c) < d(b, c)\) add \(d(a, c)\) to flip count
        - else add \(d(b, c)\) to flip count
      - end
    * set \(d(\Gamma^a, \Gamma^b) = \text{flip count}\)
    * if \(d(\Gamma^a, \Gamma^b) < \text{min}\)
      - set \(\gamma^1 = a\)
      - set \(\gamma^2 = b\)
      - set min = \(d(\Gamma^a, \Gamma^b)\)
    * end
  - end.

This algorithm enumerates each of the \(\binom{m}{2}\) possible fragment pairs in search of the optimal pair that minimizes the assigned distance. If the reader is not familiar with enumerative algorithms on networks and graphs, we recommend the following book cited in the bibliography [2]. We prove that our algorithm produces an optimal solution to \(MLF'\). Notice that in the algorithm, \(a, b,\) and \(c\) are fragments in the given haplotype set \(H\). Fragments \(a\) and \(b\) represent the current iteration’s test medians. If these medians yield a smaller flip count than what is the current minimum, they become the new \((\gamma^1, \gamma^2)\). In this manner, the \((\gamma^1, \gamma^2)\) at the end of the iterative process will have the smallest flip count and each possible pair of parent medians will have been tested. We next show that it is optimal for each fragment to be assigned to its nearest median and we call this assignment the nearest vertex assignment (partition). Then we show that the algorithm produces the same \(\{\gamma^1, \gamma^2\}\) we set out to find in our problem description.

Lemma 5.1. Let \(v^1\) be a vertex of the connected graph \(g = (V, E)\) with a 2-median solution of \((\gamma^1, \gamma^2)\). Then, the nearest vertex assignment (partition) minimizes the total distance of assigning \(v^1\) to \(\gamma^1\) or \(\gamma^2\).

Proof. Let \(w_{11}\) be the distance between \(v^1\) and its nearest median \(\gamma^1\). Since \(\gamma^1\) is its nearest median,
Lemma 5.1 states that any partition besides the nearest vertex partition does not need to be considered because it is not an optimal assignment. This allows us to focus on finding the medians rather than the partition sets because we have shown that the nearest vertex partition is optimal. From here we can make several claims about the performance of Algorithm 1.

**Theorem 5.1.** Algorithm 1 generates an optimal pair of medians for a complete graph in $O(m^3)$.

**Proof.** Let Algorithm 1 terminate with $\{\gamma^1, \gamma^2\}$. Suppose that $\{v^1, v^2\}$ is a set of medians. From Lemma 5.1 we know that the minimum assignment of $V$ to $\{v^1, v^2\}$ is the nearest vertex assignment. However, the nearest vertex assignment of $\{\gamma^1, \gamma^2\}$ has an assignment distance no greater than the nearest vertex assignment distance of assigning $V$ to $\{v^1, v^2\}$ because every possible pair of medians is considered in Algorithm 1. Hence, $\{\gamma^1, \gamma^2\}$ is an optimal assignment.

The fact that the algorithm is polynomial follows because the three for loops require $O(m^3)$ iterations, with each iteration requiring a minimum distance between two vertices. □

Now that we have shown that the $MLF'$ problem is solvable in polynomial time, we attempt to reconstruct the parent haplotypes from the $\{\gamma^1, \gamma^2\}$ that Algorithm 1 returns. As is seen in the upcoming section, many concerns surface about the optimality of the solution Algorithm 1 returns.

### 5.2 Complications

Even though Algorithm 1 successfully finds $\{\gamma^1, \gamma^2\}$ in the set of fragments, we cannot infer the unique parent haplotypes, $h^{1*}$ and $h^{2*}$, after performing the necessary number of flips Algorithm 1 returns. When we say *unique parent haplotype*, we mean that there are no '-'s in any SNP location in either $h^{1*}$ or $h^{2*}$, and there is no more than one possible $h^{1*}$ and $h^{2*}$ for a $\gamma^1$ and $\gamma^2$ returned. To see why we still cannot infer $h^{1*}$ and $h^{2*}$ after we solve the $MLF'$, we first need to introduce more notation.

After running Algorithm 1, we use the flip count returned to alter the given fragments so that the new fragments are zero distance away from their respective median. Here we need to introduce the notation $\hat{\Gamma}^{\gamma^1}$ and $\hat{\Gamma}^{\gamma^2}$. This notation is necessary because $\{\Gamma^{\gamma^1}, \Gamma^{\gamma^2}\}$ only denotes the set of fragments assigned to $\{\gamma^1, \gamma^2\}$; no SNPs have been changed to ensure that all fragments in $\Gamma^{\gamma^1}$ are zero distance away from $\gamma^1$. $\hat{\Gamma}^{\gamma^1}$ is the set of altered fragments in $\Gamma^{\gamma^1}$ after flipping has occurred. $\hat{\Gamma}^{\gamma^2}$ is defined similarly. Consider the following example:
Given: $\gamma^1 = \ldots$ABA and
\[
\Gamma^{\gamma^1} = \{-ABA, ABBBA, BBABB\}
\]
flip count = 2. Flipping the conflicting SNPs yields
\[
\hat{\Gamma}^{\gamma^1} = \{-ABA, ABABA, BBABA\}
\]
Notice that $\gamma^1$ is not $\subseteq$ to all of the fragments in $\Gamma^{\gamma^1}$. However, $\gamma^1$ is $\subseteq$ each element in $\hat{\Gamma}^{\gamma^1}$. Moreover, the flip count that Algorithm 1 returns indicates that this is the number of SNPs that have to be changed so that $\gamma^1$ $\subseteq$ each element of $\hat{\Gamma}^{\gamma^1}$ and $\gamma^2$ $\subseteq$ each element of $\hat{\Gamma}^{\gamma^2}$. It is important to understand that $\hat{\Gamma}^{\gamma^1}$ does not repartition the fragments, it is simply the set of fragments that have been corrected to become conflict-free with the nearest median.

After determining $\hat{\Gamma}^{\gamma^1}$ and $\hat{\Gamma}^{\gamma^2}$, we use a technique called overlapping in an attempt to find $h^{1*}$ and $h^{2*}$. We use the notation $h^*$ to indicate a haplotype that represents the final, completed parent haplotypes. This is not to be confused with the $\gamma^i$ notation that represents a vertex solution to a 2-median problem. Overlapping means that after we have our $\gamma^1$ we try to fill in any $-$’s by looking at $\hat{\Gamma}^{\gamma^1}$. This is a reasonable procedure because none of the known SNP locations in $\gamma^1$ can be changed since $\gamma^1$ $\sim$ all fragments in $\hat{\Gamma}^{\gamma^1}$. Moreover, filling in the $-$’s is precisely what we wish to attain since, by definition, $h^{1*}$ has no SNP location that are $-$’s. However, by considering the following example, we state the following fact:

**Fact:** After using the number of flips Algorithm 1 returns to produce $\{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^2}\}$ and overlapping the fragments, we cannot infer the complete parent haplotypes, $h^{1*}$ and $h^{2*}$.

**Example 5.1.** Consider the following matrix,

\[
\begin{array}{ccc}
A & B & - \\
B & B & A \\
- & A & B \\
- & - & A & B
\end{array}
\]

For this matrix, Algorithm 1 would return that $\{\gamma^1, \gamma^2\} = \{-ABA, -AB\}$ with the number of flips required equal to 0. In this matrix $\Gamma^{\gamma^1} = \{-ABA\}$ and $\Gamma^{\gamma^2} = \{AB-, BBA-, -AB\}$. Since the flip count = 0, $\{\Gamma^{\gamma^1}, \Gamma^{\gamma^2}\} = \{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^2}\}$. In this case, overlapping completes the – in the second SNP in $\gamma^2$, but does not complete the – in the first SNP position in $\gamma^2$. $h^{2*}$ could either be $BBAB$ or $ABAB$. However, the definition of $h^{2*}$ is that it has to be unique for a given $\gamma^2$. For this reason, we say that we cannot infer the complete parent haplotypes, $h^{1*}$ and $h^{2*}$, after performing the necessary number of flips returned by Algorithm 1.

Since we cannot directly infer $h^{1*}$ and $h^{2*}$ from MLF*, our original intuition was that it might be possible to use the number of flips that Algorithm 1 returns, construct $\{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^2}\}$, overlap the fragments, and remove the $-$’s in $\{\gamma^1, \gamma^2\}$ such that $\{\gamma^1, \gamma^2\}$ are the solutions to the CGMLF. Here is an example:
Consider the matrix from Example 5.1. Since we have already noticed that it is impossible to infer $h^{1*}$ and $h^{2*}$ in 0 flips, we note that one optimal solution to CGMLF is $\{AABA, ABAA\}$ since it only requires one flip (changing the first SNP in the second fragment to an A). Our $\{\gamma^1, \gamma^2\} = \{-ABA, -AA\}$ that Algorithm 1 returned is indeed $\sim$ of this optimal solution $\{h^{1*}, h^{2*}\} = \{AABA, ABAA\}$. Note that there exist other optimal solutions to CGMLF, but all that is required is to show that our $\{\gamma^1, \gamma^2\} \sim \sim$ of one of them.

However, contrary to our initial conjecture, we have shown by counterexample the following fact:

**Fact:** The $\{\gamma^1, \gamma^2\}$ that Algorithm 1 returns is not always $\subseteq$ of the solutions to CGMLF.

The following example illustrated two cases in which $\{\gamma^1, \gamma^2\}$ can fail to be $\subseteq$ of the solutions to CGMLF.

**Example 5.2.** Case 1: The medians of the MLF are not $\subseteq$ of the solutions to CGMLF. Below is a matrix with fragments whose $\{\gamma^1, \gamma^2\}$ not $\sim \sim \subseteq \{h^{1*}, h^{2*}\}$.

$$
\begin{array}{ccc}
B & - & A \\
A & A & A \\
- & B & A \\
A & - & A \\
* & - & B \\
* & - & A \\
\end{array}
$$

The * fragments indicate the $\{\gamma^1, \gamma^2\}$ that Algorithm 1 returns. Based on Algorithm 1, the assignments in this matrix would be $\Gamma^{\gamma^1} = \{-BB\}$ and $\Gamma^{\gamma^2} = \{-AA, B-AA, AAA-\}$, so $\{h^{1*}, h^{2*}\} = \{AAAB, AAAAA\}$. However, suppose we let $\{h^{1*}, h^{2*}\} = \{AAAB, BBAA\}$ and make the following assignments, $\Gamma^{\gamma^1} = \{-BB, AAA-, A-A-\}$ and $\Gamma^{\gamma^2} = \{-AA, B-AA, -BAA\}$. It would only require flipping the third SNP in the fifth fragment to an A and overlap the rest with no penalty to get $\{h^{1*}, h^{2*}\} = \{AAAB, BBAA\}$. However, $\{\gamma^1, \gamma^2\} = \{-BB, -AA\}$ returned by Algorithm 1 is not $\subseteq$ of this optimal solution CGMLF = $\{AAAB, BBAA\}$, which only takes one flip.

Case 2: The second case in which Algorithm 1 fails to give a $\{\gamma^1, \gamma^2\}$ that is $\subseteq$ of the solutions to CGMLF is the following:
Here, \( \{\gamma^1, \gamma^2\} = \{AA\ldots, \ldots AA\} \), and the flip count Algorithm 1 returns 0. Producing a \( h^1* \) and a \( h^2* \) by overlapping \( \Gamma^{-\gamma^1} \) and \( \Gamma^{-\gamma^2} \) requires 4 flips. However, suppose we let \( CGMLF = \{BBBBBBBBB,AAAAAAAAA\} \). It only takes 3 flips to construct \( CGMLF \) solution from \( H \), and \( \{\gamma^1, \gamma^2\} \) is not a subset of these \( CGMLF \) solutions.

What is apparent from the previous examples is that the \( MLF' \) is not the exact problem we wish to approach. The biggest flaw in the \( MLF' \) problem is that it does not produce a bipartite graph. This is a problem because in the real \( CGMLF \) problem statement, a bipartite graph is required. What follows is an example that illustrates why \( MLF' \) does not produce a bipartite graph:

**Example 5.3.** Consider the following simple SNP matrix:

\[
\begin{array}{ccc}
A & B & - \\
B & B & A \\
- & A & B \\
- & A & B \\
\end{array}
\]

\( \gamma^1 = ABA \), \( \gamma^2 = AB \).

\( \Gamma^{-\gamma^1} = \{ABA\} \).

\( \Gamma^{-\gamma^2} = \{AB, BBA, AB\} \).

**flip count = 0.**

The problem is that the \( \Gamma^{-\gamma^2} \) partition is not conflict-free because even though \( AB \) and \( BBA \) are zero distance away from their median \( AB \), they are not zero distance away from each other (i.e., there is an edge between two members of the same partition.) Hence \( MLF' \) does not always produce a bipartite graph.

Since we now understand that \( MLF' \) does not solve the \( CGMLF \), we consider \( DMLF \) and \( DMLF' \). In the following sections, we discuss an algorithm, results, and how \( DMLF \) and \( DMLF' \) relate to \( CGMLF \).

### 6 DMLF' and DMLF

To reintroduce \( DMLF' \), it is similar to the \( MLF' \) problem formulation with the exception that it uses the \( l \)-distance measure instead of the \( d \)-distance. Recall that the \( l \)-distance is not symmetric.
and penalizes a flip from a known SNP to an unknown –. The DMLF that we discuss in this section is similar to the DMLF', except that we are not restricted to search for medians that are inside our given set of fragments, $H$. The following is an optimal algorithm to solve DMLF'. The subsection that proceeds Algorithm 2 discusses some crucial properties and results for the DMLF and the DMLF'.

6.1 DMLF' Algorithm

The algorithm presented here determines both the minimal number of flips and a $\{\gamma^1, \gamma^2\}$ that solves the DMLF' problem.

**ALGORITHM 2**

Step 0: Set $\text{min} = \infty$.

Step 1: Arbitrarily number the fragments 1 through $m$.

Step 2: 
- for $(a = 1, m)$
  - for $(b = 1, m)$
    - flip count $= 0$
      - for $(c = 1, m)$
        - if $l(a, c) < l(b, c)$ add $l(a, c)$ to flip count
        - else add $l(b, c)$ to flip count
      - end
    - set $l(\Gamma^a, \Gamma^b) = \text{flip count}$
    - if $l(\Gamma^a, \Gamma^b) < \text{min}$
      - set $\gamma^1 = a$
      - set $\gamma^2 = b$
      - set $\text{min} = l(\Gamma^a, \Gamma^b)$
    - end
  - end
- end.

Algorithm 2 finds the directed distance for each of the $2 \times \binom{m}{2}$ possible fragment pairs. There are twice as many possible combinations because we have a distinction between $l(v^i, v^j)$ and $l(v^j, v^i)$ since their distances may not be the same. Algorithm 1, on the other hand, always considers $d(v^i, v^j)$ and $d(v^j, v^i)$ to be equal. The process is similar to Algorithm 1, but with an extended range in the second for loop (to get a distance for both $l(v^i, v^j)$ and $l(v^j, v^i)$) and the use of the $l$-distance to determine the weight.
6.2 Results

Due to the similarity of Algorithm 1 to Algorithm 2, we can use Lemma 5.1 and Theorem 5.1 (which deal with the optimality of the Algorithm 1) to extend and apply to Algorithm 2.

**Theorem 6.1.** Algorithm 2 generates an optimal pair of medians for DMLF'.

*Proof.* Let Algorithm 2 terminate with \{γ^1, γ^2\}. Suppose that \{v^1, v^2\} is a set of medians. From Lemma 5.1 we know that the minimum assignment of V to \{v^1, v^2\} is the nearest vertex assignment. However, the nearest vertex assignment of \{γ^1, γ^2\} has an assignment distance no greater than the nearest vertex assignment distance of assigning V to \{v^1, v^2\} because every possible pair of medians is considered in Algorithm 2. Hence, \{γ^1, γ^2\} is an optimal assignment. □

Similar to the previous section on MLF', the following is a proof that DMLF' is polynomial.

**Theorem 6.2.** Algorithm 2 is O(m^3).

*Proof.* To show that the Algorithm 2 is O(m^3), we consider that it enumerates through each possible pair of fragments. To create each pair of fragments, there are 2× \binom{m}{2} = 2× \frac{m!}{(m-2)!2!} = \frac{m(m-1)}{2} = (m^2 - m) possible combinations of m fragments. Each of these pairs is then compared against all other fragments in H to find each minimum distance assignment, which requires 2m steps. This results in an algorithm that is O(2m^3 - 2m^2), which is O(m^3). □

Now that we know that DMLF' can be solved in polynomial time, we consider the complexity of DMLF. In Theorem 6.3, we see that DMLF is solvable in exponential time.

**Theorem 6.3.** Algorithm 2 applied to DMLF is O(3^n)

*Proof.* Recall from Theorem 6.2, that we can solve the 2-median problem over m fragments in O(m^3). Since DMLF can choose medians from any S_n = \{A, B, \}^n, we have m = 3^n fragments to select from. This implies that using Algorithm 2 to solve DMLF is O((3^n)^3) = O(3^{3n}). □

Although using Algorithm 2 to solve DMLF is not polynomial, we can test the proposed solution in a polynomial number of steps. Generating the 3^n possibilities for fragments is what makes this algorithm non-polynomial; however, simply testing the 2-median solution can be completed in polynomial time.

Although we have shown the enumerative algorithm and explained the l-distance for DMLF' and DMLF, it is not immediately obvious what solving these problems accomplishes. Because DMLF will have an important role in the upcoming section, the rest of this section is devoted to an informal explanation of exactly how DMLF' and DMLF work.

When DMLF' and DMLF select medians, they will inherently prefer a non—SNP to a —. The following is an explanation of why.
Consider a fragment $h^i=AB-$ that is assigned to some median. This median could be $-\ldots-$, in which case the distance from $h^i$ to the median is 2. In other words, there is no benefit of having all $-\ldots-$ as a median; Algorithm 2 never picks this solution. Now suppose that the median is $AB-$, which is a good solution since the distance is zero. However, ABA and ABB are equally good solutions (flip count 0) and all SNP positions are filled. Moreover, when the flip count for a letter is equal to the flip count for a $-$, Algorithm 2 will choose the letter.

Once Algorithm 2 picks out a $\{\gamma^1, \gamma^2\}$, it assigns the fragments in $H$ to their nearest median to form $\{\Gamma^\gamma_1, \Gamma^\gamma_2\}$. From the flip count Algorithm 2 returns, we then form $\{\hat{\Gamma}^\gamma_1, \hat{\Gamma}^\gamma_2\}$. Remember that $\{\hat{\Gamma}^\gamma_1, \hat{\Gamma}^\gamma_2\}$ denotes the altered sets of fragments in each partition ($\hat{h}^i \in \hat{\Gamma}^\gamma_1$ iff $h^i \in \Gamma^\gamma_1$, $\hat{h}^j \in \hat{\Gamma}^\gamma_2$ iff $h^j \in \Gamma^\gamma_2$) such that the distance between each fragment in a given partition and its nearest median is zero. Because the Algorithm is using $l$-distance, it is not possible to have a $-$ in the $k^{th}$ SNP position of $\gamma^1$ where there is no dash in the $k^{th}$ SNP position of $\hat{h}^i$ (since the distance from $h^i$ to $\gamma^1$ would not be zero) for all SNP positions $k$. Below is an example to illustrate this point:

Suppose $\gamma^1=AB-$.
By definition of $\hat{\Gamma}^\gamma_1$, all fragments $\hat{h}^i \in \hat{\Gamma}^\gamma_1$ must be zero distance away from $\gamma^1$.
Therefore, $\hat{h}^i$ could not have a defined SNP in the third position since $l(\hat{h}^i, \gamma^1)$ would not be zero.

Since no $\hat{h}^i \in \hat{\Gamma}^\gamma_1$ can have a defined SNP position where $\gamma^1$ is undefined, the following property about $DMLF^r$ and $DMLF$ falls out nicely:

**Property 1:** Given $\hat{h}^i \in \hat{\Gamma}^\gamma_1$, $\hat{h}^i \sim \gamma^1 \forall h^i \in \Gamma^\gamma_1$. Similarly, given $\hat{h}^j \in \hat{\Gamma}^\gamma_2$, $\hat{h}^j \sim \gamma^2 \forall h^j \in \Gamma^\gamma_2$.

Another important observation about $DMLF^r$ and $DMLF$ is how we build $\hat{h}$. Even though we know that $\hat{h}^i \sim \gamma^1 \forall h^i \in \Gamma^\gamma_1$, there are several options for what $\hat{h}^i$ could be. For example, suppose that $\gamma^1=AB-$, and $h^i \in \Gamma^\gamma_1$ and $h^1=BB-$. Two possibilities for $\hat{h}^i$ are $-B-$ and $AB-$. In this case $l(h^i, \hat{h}^i)$, where $\hat{h}^i = -B-$, is 1. Likewise, $l(h^i, \hat{h}^i)$, where $\hat{h}^i = AB-$, is 1. Since $\hat{h}^i = AB-$ takes the same number of flips as $\hat{h}^i = -B-$, we will always construct $\hat{h}^i$ in such a way that we have as many defined SNP positions as possible; in this case we would pick $\hat{h}^i = AB-$.

The reason we spend so much time explaining these facts and properties of $DMLF$ is because intuitively, by Property 1, $DMLF$ will induce a bipartite graph (in which case it can be compared to $CGMLF$). The next section is devoted to precisely this idea.
7 Relating DMLF to CGMLF

The DMLF problem is the most biologically relevant of the 2-median problem formulations. As stated earlier, it uses the $l$-distance, which forces the medians to have as many defined SNPs as possible in the medians. Since the DMLF can locate the medians over all $S_n$, the two medians it selects are defined in every location where possible, resulting in accurate parent haplotypes. In this section, we show that solving DMLF, the problem of finding feasible partitions, is equivalent to solving CGMLF.

To re-enforce our notation, when we refer to the 2-median problems (DMLF in this case) we use $\{\gamma^1, \gamma^2\}$ as our 2-median solutions. When we talk about the CG(H) problems (CGMLF in this case), we use $\{s^*, s^t\}$ as the generated medians. In other words, $\{\gamma^1, \gamma^2\}$ are haplotypes the algorithm returns while $\{s^*, s^t\}$ are haplotypes that can be constructed from the partitions of CGMLF. Similarly, $\{\Gamma^r, \Gamma^s\}$ and $\{\Gamma^r', \Gamma^s'\}$ are used for the 2-median problems, while $\{H^1, H^2\}$ and $\{\bar{H}^1, \bar{H}^2\}$ are used for the CG(H) problems.

The following Lemma will allow us to generalize an important property of a feasible partition. This lays the groundwork for showing that the DMLF is equivalent to solving the CGMLF.

**Lemma 7.1.** CG(H) is bipartite $\iff \exists$ a partition of $H$, say $\{H^1, H^2\}$, such that every SNP in a partition set is either in $\{A, -\}$, $\{B, -\}$, or is all $-$'s.

**Proof.**

**Part 1:** If CG(H) is bipartite $\Rightarrow$ every SNP in a partition set is either in $\{A, -\}$, $\{B, -\}$, or just $\{-\}$.

By definition of a bipartite graph, $\forall h^i, h^j \in H^i$, where $H^i$ is any partition of CG(H), $h^i$ does not conflict with $h^j$. This means that $h^i_k$ does not conflict with $h^j_k \ \forall \ k \in \{1...n\}$. Moreover, there is no $k$ for which $h^i_k=A$ and $h^j_k=B$. Therefore, every SNP in a partition set is either in $\{A, -\}$, $\{B, -\}$, or $\{-\}$.

**Part 2:** If every SNP in a partition set is either in $\{A, -\}$, $\{B, -\}$, or just $\{-\}$, $\Rightarrow$ CG(H) is bipartite.

By definition of two SNPs being in conflict, if every SNP in a partition set is either in $\{A, -\}$, $\{B, -\}$, or is just $\{-\}$, there does not exist a $k$, such that $h^i_k, h^j_k \in H^i$ conflict. This implies that $h^i$ and $h^j$ do not conflict $\forall h^i, h^j \in H^i$. The same argument is applied to any partition set $H^i$. Since for all partition sets in $H$, there are no fragments that conflict $\forall h^i, h^j \in H^i$, CG(H) is bipartite.

Lemma 7.1 states that a feasible partition will look similar to the example in Table 3, where each SNP column will consist of only $\{A, -\}$, $\{B, -\}$, or $\{-\}$. It is clear to see that a $B$ in column 1 would conflict with the A SNPs and the partition would not be bipartite. Likewise, an A in a $\{B, -\}$ column would not be conflict-free either. It is important to remember that some letters may have been flipped in order to create the feasible partition. The following rule applies to generating a parent haplotype using this existing property of feasible partitions. This rule will be used to later
Table 3: An example of a feasible partition

<table>
<thead>
<tr>
<th>SNP</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>-</td>
<td>A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>A</td>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>B</td>
<td>-</td>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>-</td>
<td>B</td>
</tr>
</tbody>
</table>

show that the partitions from the CGMLF are equivalent to the partitions of the DMLF.

**Rule 1:** Generating a parent haplotype from a partition of a bipartite collection of haplotypes. By definition of a bipartite subgraph, for $i \in \{1, 2\}$, we have that $\forall \hat{h}_k \in \hat{H}^i$, $\hat{h}_k \in \{A, -\}$ or $\hat{h}_k \in \{B, -\}$. To generate a parent haplotype, we define $s^1_k$ by:

$$s^1_k = \begin{cases} A, & \text{if } \hat{h}_k \in \{A, -\} \text{ for all } \hat{h} \in \hat{H}^1 \\ B, & \text{if } \hat{h}_k \in \{B, -\} \text{ for all } \hat{h} \in \hat{H}^1 \\ - , & \text{if } \hat{h}_k \in \{-\} \text{ for all } \hat{h} \in \hat{H}^1 , \end{cases}$$

for $k \in \{1...n\}$. We define $s^2_k$ in the same manner, where $\hat{h} \in \hat{H}^2$ instead of $\hat{h} \in \hat{H}^1$.

Looking at Table 3, applying Rule 1 to this partition would yield a parent haplotype of $\{ABA - B\}$. Note that the only time $-$ will be returned by Rule 1 is when a column consists only of ambiguous SNPs. This is unlikely biologically because it would mean that the sequencer could not call with certainty a SNP from any fragment derived from a parental donation. Now that we have two important concepts formulated, we show that CGMLF is equivalent to DMLF because $z(DMLF) = z(CGMLF)$, that is, their flip counts are the same.

**Theorem 7.1.** $z(DMLF) = z(CGMLF)$

**Proof.** There are two parts to this proof. First, we show that DMLF induces a bipartite graph, which implies that $z(DMLF) \geq z(CGMLF)$. This follows because CGMLF finds a bipartite graph with the minimum number of flips, and we cannot yet say that DMLF produces this optimal bipartite graph. The second part of the proof (by contradiction) shows that DMLF is optimal, implying that $z(DMLF) = z(CGMLF)$.

**Part 1:** DMLF makes a bipartite graph.

From Property 1 of DMLF, we know that $\hat{h}^i \sim \gamma \forall \hat{h}^i \in \Gamma$. By the definition of $\sim$, $\hat{h}^i \sim \gamma$ iff
$h'_k = \gamma_k$ or $h'_k = -\forall k$. Since $\hat{h} \subseteq \gamma_1$, any $\hat{h}_k \in \hat{H}^k$ is either in $\{A,-\}$, $\{B,-\}$, or just $\{-\}$. The same argument holds for all $\hat{h} \in \hat{H}^2$. By Lemma 7.1, if $\exists$ a partition of $H (\{\hat{\gamma}_1, \hat{\gamma}_2\})$ such that every SNP partition set is either in $\{A,-\}$, $\{B,-\}$, or $\{-\}$, then $CG(H)$ is bipartite. Thus, $DMLF$ produces a bipartite graph and we say $z(DMLF) \geq z(CGMLF)$.

**Part 2: DMLF is the optimal bipartite graph.**

Assume $DMLF$ does not induce the optimal bipartite graph. Then, $z(CGMLF) < z(DMLF)$ for the same matrix $H$. Let $\{\hat{H}_1, \hat{H}_2\}$ be a partition of $CG(\hat{H})$ such that $d(H, \hat{H}_1 \cup \hat{H}_2) = z(CGMLF)$. This notation comes from the definition of $CGMLF$; the distance from the set of fragments, $H$, to the two bipartite partitions $\{\hat{H}_1 \cup \hat{H}_2\}$ is precisely what is meant by the flip count of $CGMLF$.

Since the partitions make a bipartite graph, the $k^{th}$ SNP for every $\hat{h} \in \hat{H}_1$ is either in $\{A,-\}$ or $\{B,-\} \forall k$. Following Rule 1, let

$$s^1_k = \begin{cases} A, & \text{if } \hat{h}_k \in \{A,-\} \text{ for all } \hat{h} \in \hat{H}_1 \\ B, & \text{if } \hat{h}_k \in \{B,-\} \text{ for all } \hat{h} \in \hat{H}_1 \\ - & \text{if } \hat{h}_k \in \{-\} \text{ for all } \hat{h} \in \hat{H}_1. \end{cases}$$

By doing this, we have generated $s^1$ from $\hat{H}_1$. Similarly generate $s^2$ from $\hat{H}_2$.

From our initial statement that $d(H, \hat{H}_1 \cup \hat{H}_2) = z(CGMLF)$, we know that

$$z(CGMLF) = \sum_{\substack{j \in J \in H}} d(h'_j, \hat{h}'_j) = \sum_{\substack{j \in J \in H}} d(h'_j, \hat{h}'_j) \cup \sum_{\substack{j \in J \in H^2}} d(h'_j, \hat{h}'_j). \quad (1)$$

Since the partitions are bipartite by definition of $CGMLF$, $\hat{H}_1 \cap \hat{H}_2 = \emptyset$. We now write,

$$z(CGMLF) = \sum_{\substack{j \in J \in H}} d(h'_j, \hat{h}'_j) = \sum_{\substack{j \in J \in H^1}} d(h'_j, \hat{h}'_j) + \sum_{\substack{j \in J \in H^2}} d(h'_j, \hat{h}'_j). \quad (2)$$

Now that we know what $z(CGMLF)$ is, we choose the same $\{s^1, s^2\}$ that we just generated as the medians with optimal flip count from $CGMLF$ to be the $\{\gamma^1, \gamma^2\}$ for $DMLF$. Note that we are not saying that these $\{s^1, s^2\}$ are the optimal haplotypes, $\{\gamma^1, \gamma^2\}$, that $DMLF$ would have produced on its own by running Algorithm 2. Moreover, we will show that the flip count $z(DMLF)$ using $\{s^1, s^2\}$ is not greater than $z(CGMLF)$, giving us a contradiction. Since we have chosen $\{s^1, s^2\}$ to be the $\{\gamma^1, \gamma^2\}$, $DMLF$ will assign each $h'$ to its nearest median and ultimately partition $H$ into bipartite $\{\Gamma^1, \Gamma^2\}$. In other words, $h'$ takes some number of flips to become $\hat{h}'$ such that $\hat{h}' \subseteq s^r$ where $r \in 1, 2$. This point is reiterated with the following statement.

$$z(DMLF) = \sum_{\substack{j \in J \in H^1}} w_{i,s^1} + \sum_{\substack{j \in J \in H^2}} w_{i,s^2} = \sum_{\substack{j \in J \in H^1}} l(h'_j, s^1_j) + \sum_{\substack{j \in J \in H^2}} l(h'_j, s^2_j) + \sum_{\substack{j \in J \in H^1}} (h'_j, s^1_j) + \sum_{\substack{j \in J \in H^2}} (h'_j, s^2_j). \quad (4)$$
However, by definition of $h^r$, any $h^r$ is conflict-free with the parent haplotype, so $\sum_\limits_{h^i \in G^2} d(h^i_j, s_j^1)$ and $\sum_\limits_{h^i \in G^2} d(h^i_j, s_j^2)$ are 0. Therefore,

$$z(DMLF) = \sum_\limits_{h^i \in G^1} l(h^i_j, \hat{h}^i_j) + \sum_\limits_{h^i \in G^2} l(h^i_j, \hat{h}^i_j).$$  \hspace{1cm} (5)$$

Next, recall that the set of fragments $H$ and the medians $\{s^1, s^2\}$ are the same in $CGMLF$ as in $DMLF$. This implies that $\{G^1, G^2\}$ in $DMLF$ is the same as $\{H^1, H^2\}$ in $CGMLF$. Moreover, the most important step is to see that the distance $l$ we are using for $z(DMLF)$ is equivalent to the $d$-distance for the following reason:

When we are counting the minimum number of flips from a given $h^i \in H^1$ to a $\hat{h}^i$ such that $\hat{h}^i \sim s^r$ where $r \in 1, 2$, given any SNP position $k \in h^i$, we never go from an A to a $-$ or a B to a $-$. The only reason to change an A or B to a $-$ at a given SNP position is if $s^r$ has a $-$ in that SNP position. However, by Rule 1, for $s^r$ to have had a $-$ in that SNP position, every $h^i \in H^1$ must have had a $-$ in that SNP position as well (in which case what we are trying to turn into a $-$ would have already been a $-$). Since we would never go from an A to a $-$, or from a B to a $-$, $l$-distance is identical to $d$-distance.

For this reason, Equation 5 becomes:

$$z(DMLF) = \sum_\limits_{h^i \in H^1} d(h^i_j, \hat{h}^i_j) + \sum_\limits_{h^i \in H^2} d(h^i_j, \hat{h}^i_j).$$  \hspace{1cm} (6)$$

However, Equation 2, which gives $z(CGMLF)$ is identical to Equation 6, which gives us $z(DMLF)$. This is a contradiction to our assumption that $z(CGMLF) < z(DMLF)$.

Moreover, since we know from Part 1 of this proof that $z(DMLF) \geq z(CGMLF)$, and from Part 2 that $z(DMLF) > z(CGMLF)$ is not true, we have proven that $z(DMLF) = z(CGMLF)$. \hspace{1cm} \square

This theorem states that solving $CGMLF$ takes the same number of flips as solving $DMLF$. This is an important result, because we have now related the original problem of locating feasible partitions to a 2-median problem.

$z(CGMLF) = z(DMLF)$ is a significant result because we have a lot of results in place referring to $DMLF$. From the previous section, we showed that $DMLF$ can be solved in non-polynomial time, which means we can solve $CGMLF$ in non-polynomial time as well. Since we have built the
relation between \( CGMLF \) and \( DMLF \), in the following section we attempt to construct relations between the remaining problem statements.

8 Relating Each Problem Formulation

In this section, we explore the relationship between \( MLF \), \( MLF' \), \( DMLF \), \( DMLF' \) and \( CGMLF \) in terms of which problem statement requires the minimum number of flips. For simplicity's sake, we say problem statement \( I \) is \( < \) problem statement \( J \) if \( z(I) \leq z(J) \). Before we can create the string of inequalities, we need to consider the following two lemmas.

Lemma 8.1. Given a 2-median problem statement \( I \) and a 2-median problem statement \( J \), where the distance measures are identical but problem statement \( I \) finds medians in \( S_n \) and \( J \) finds medians in \( H \), then \( I \leq J \).

**Proof.** This lemma is true by the following optimization result: \( \min\{f(x) : x \in X\} \leq \min\{f(x) : x \in Y \subseteq X\} \) because \( H \subseteq S_n \). □

Lemma 8.2. Given a 2-median problem statement \( I \) and a 2-median problem statement \( J \), where both \( I \) and \( J \) find medians from the same set but problem statement \( I \) uses \( d \)-distance and \( J \) uses \( l \)-distance, then \( I \leq J \).

**Proof.** This lemma is true by the following optimization result: \( \min\{(c^1)^T x : x \in X\} \leq \min\{(c^2)^T x : x \in X\}, \) where \( 0 \leq c^1 \leq c^2 \)\( Vi. \) In this case, \( c^1 \) represents the \( d \)-distance, while \( c^2 \) is the \( l \)-distance. □

With the help of these lemmas, we have the following theorem:

Theorem 8.1. \( MLF \leq MLF' \leq DMLF' \) and \( MLF \leq DMLF = CGMLF \leq DMLF' \).

**Proof.** 1) \( MLF \leq MLF' \leq DMLF' \).

\( MLF \leq MLF' \) from Lemma 8.1 and \( MLF' \leq DMLF' \) from Lemma 8.2. Thus, \( MLF \leq MLF' \leq DMLF' \).

2) \( MLF \leq DMLF = CGMLF \leq DMLF' \).

\( MLF \leq DMLF \) from Lemma 8.2 and \( DMLF \leq DMLF' \) from Lemma 8.1. In addition, from Theorem 7.1, \( DMLF = CGMLF \). Thus, \( MLF \leq DMLF = CGMLF \leq DMLF' \). Therefore, \( MLF \leq MLF' \leq DMLF' \) and \( MLF \leq DMLF = CGMLF \leq DMLF' \). □

It would be helpful if we could form a single system of inequalities, but first we need to find a relation between \( MLF' \) and \( DMLF \). The reason this is not easily accomplished is because \( DMLF \) searches through \( S_n \) and \( MLF' \) searches through \( H \), intuitively implying that \( DMLF \) should be \( \leq MLF' \). However, Lemma 8.1 doesn't apply since these problems use different distance measures. The following SNP matrix illustrates this point:
Here, $MLF'$ would determine that 0 flips are required with $\gamma^1 = AA$ and $\Gamma^2 = \{AA, -AA, -AB\}$ while $BB -$ is the only element in $\Gamma^2$ and is therefore $\gamma^2$. However, $DMLF$ would determine that 1 flip is required and there are several solutions that fulfill this. This flip will occur when changing the $-AB$ fragment where conflict exists with either the A or the B SNP and the other fragments in the set. Therefore, we can conclude that $DMLF$ is not always $\leq MLF'$.

Since we have not found an example of where $DMLF \leq MLF'$, we believe that $MLF' \leq DMLF$. This becomes difficult to prove because these two problem formulations are so inherently different. Both their median search sets and distance measures are different and they may produce fragment partitions and median solutions that aren't the same. Our inability to find this relation prohibits us from saying that $CGMLF$ is bounded from below by a polynomial 2-median problem. However, since we have proven that $CGMLF \leq DMLF'$, we can already say that $CGMLF$ is bounded from above by a polynomial 2-median problem. This means an upper bound can be determined in polynomial time.

9 Conclusion and Future Work

There are several approaches available to minimize the number of letters flips to make a SNP matrix feasible. We refer to the $CGMLF$ as the problem of changing a minimal number of SNPs to create two feasible partitions of the SNP matrix. Since finding partitions that have this property is especially difficult, we relate this problem to several 2-median problem formulations. Our major result is that the $CGMLF$ is equivalent to the $DMLF$, a 2-median problem using the 1-distance over the complete set of fragments $S_n$. We can bound the $DMLF$ from below by the $DMLF'$ in $O(m^3)$ time. We have further shown the minimal flip counts of the various problem formulations to be related in the following way: $MLF \leq MLF' \leq DMLF'$ and $MLF \leq DMLF = CGMLF \leq DMLF'$.

9.1 Future Work

In future work, we would form a single system of inequalities by combining the two existing systems. We conjecture that the $MLF' \leq DMLF$, but this has been shown to be difficult to prove. This would lead us to complete the system of inequalities to say that $MLF \leq MLF' \leq CGMLF = DMLF \leq DMLF'$. If this conjecture is shown to be true, the $DMLF$ and $CGMLF$ would then be bounded above and below by two polynomial time problems. In the case where $MLF'$ and $DMLF'$ are equal, we would have the $CGMLF$ and $DMLF$ flip counts in polynomial time.
Other future work would include investigating the type of sequencing errors that cause the \textit{MLF}' and \textit{DMLF}' to fail when determining parent haplotypes. If we can determine what type of errors are anticipated from the sequencer, various techniques may be developed that are more applicable to correct these expected errors.

With developments now made in the Minimum Fragment Removal and Minimum SNP Removal problems stated in the introduction, combining these error correcting techniques with the \textit{MLF} into an optimal approach is even more useful. Being able to choose which fragments to remove, SNPs to remove, and letters to flip will provide useful techniques for geneticists to apply when sequencing individuals.
References

