The Minimum Letter Flip Problem for Haplotyping a Single Individual

John Louie, Lena Sherbakov
Department of Mathematics
Trinity University
San Antonio, TX 78212.

July 30, 2004

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 $CGMLF$ is bounded from above in polynomial time.

1 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.

When a DNA strand is sequenced, a string of A's, T's, G's, and C's is generated. 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 sequencing fragments are 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.

1.1 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. 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 are suggested, ranging from the minimum fragment removal (MFR), minimum SNP removal (MSR), and the problem we investigate, the minimum letter flip (MLF) problem citeJ.

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 can only have two 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

		SNP			
		1	2	3	4
Fragments	1	A	-	-	B
	2	B	A	B	-
	3	-	A	A	-
	4	B	B	A	-
	5	A	A	-	A
	6	-	-	B	A

Table 1: SNP Matrix

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.

A 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. 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 their 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 fragment 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 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.

The problem we investigate is the MLF problem. Given a SNP matrix, we want to flip

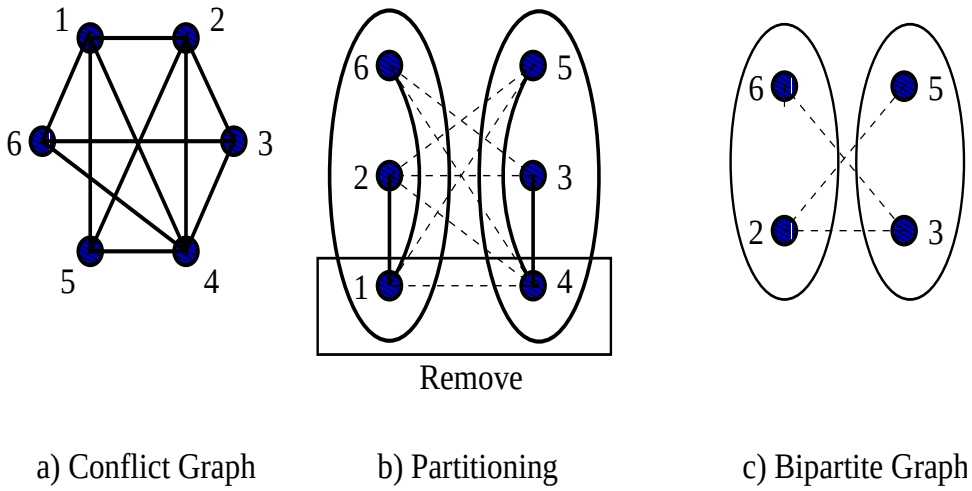

Figure 1: Conflict and Bipartite Graphs

(change) the minimum number of letters 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, since all other terms of either function can be disregarded due to the constant c term. 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.

2 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 formally define the $CGMLF$ as follows:

Definition 2.1. *The $CGMLF$ problem is a relabelling problem that transforms $H \rightarrow \hat{H} \ni CG(\hat{H})$ is bipartite and $z(H, \hat{H})$ is as small as possible.*

A *relabelling* of an $h^i \in H$ to an $\hat{h}^i$ is used when a flip may or may not have changed the representation of a fragment h^i. The $z(H, \hat{H})$ in the definition is the number of total flips required to create the feasible partition set $\hat{H}$ from the set H. Since the $CGMLF$ is a partioning problem, 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.

2.1 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, the p facilities are the two optimal haplotypes from a set of fragments and the distance between clients is specific to the problem formulation.

First, given our definition of a SNP and a fragment, we take m fragments with n SNP entries in $\{A, B, -\}$ and construct a $m \times n$ 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 $\begin{pmatrix} m \\ 2 \end{pmatrix}$ possible pairs of fragments a distance. 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 with no penalty paid for going from a $-$ to a letter or from a letter to a $-$. This distance is defined as,

$$d(h^i, h^j) = \sum_{p=1}^{n} d(h_p^i, h_p^j), \text{ where } d(h_p^i, h_p^j) = \begin{cases} 1, \text{ if } h_p^i = A, h_p^j = B \\ 1, \text{ if } h_p^i = B, h_p^j = A \\ 0, \textit{ otherwise.} \end{cases}$$

We will call this distance the d-distance. As an example, the 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 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 informally is defined to be $d(x,y)+d(y,z) \geq d(x,z)$.

The second formulation of a distance measure is

$$l(h^i, h^j) = \sum_{p=1}^{n} l(h_p^i, h_p^j), \text{ where } l(h_p^i, h_p^j) = \begin{cases} 1, \text{ if } h_p^i = A, h_p^j = B \\ 1, \text{ if } h_p^i = B, h_p^j = A \\ 1, \text{ if } h_p^i \in \{A, B\}, h_p^j = - \\ 0, \textit{ otherwise.} \end{cases}$$

We call this distance the 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 last). 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 $l(y,x)$.

These formulas apply to fragments outside of H as well. In many of the problem, 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:

$$s^i = s^j \text{ iff } s_i^k = s_j^k, \forall k.$$
$$s^i \underset{\sim}{\subset} s^j \text{ iff } s_i^k = s_j^k \text{ or } s_i^k = - \ \forall k.$$

An example of the relation $\underset{\sim}{\subset}$ is the following:

> Consider h^i= ABBA−, h^j= A−B−−, and h^k= AB−AA.
> We say that $h^j \underset{\sim}{\subset} h^i$, but h^k is not $\underset{\sim}{\subset}$ h^i because the last SNP in h^k is defined.

Another important set to define is,

$$S_n{}' = \{(s_1, s_2, ..., s_n):\ s_i \in \{A, B\}\} = \{A, B\}^n.$$

This is the set of all fragments with n SNPs, where each SNP is either an A or B.

After assigning a distance between all fragments, we look for a pair of fragments, (γ^i, γ^j), that represent one solution to the 2-median problem. Alternatively, we think of (γ^i, γ^j) as parent haplotypes. Because we only consider the 2-median problem, we denote (γ^i, γ^j) as (γ^1, γ^2) which may not necessarily be unique due to equivalent 2-median solutions. Note that this notation does not imply that γ^1 is the first fragment in H. After having located (γ^1, γ^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, Γ^{γ^1} is the set of all fragments whose nearest median is γ^i. With these new definitions, we now reformulate the p-median problem as follows:

Definition 2.2. ***General P-Median Problem on a Complete Graph:***
Let K_V be the complete graph on the nodes of V, and $U \subseteq V$. Let $E(U, V) = \{(v^i, v^j) : v^i \in U, v_j \in V\}$, where each edge is assigned a weight $w_{i,j}$ The general p-median problem is to find $M = \{\gamma^i : i \in I, |I| = p\} \subset V$ and $P = \{\Gamma^{\gamma^i} : i \in I\}$ such that:
1) $\bigcup_i \Gamma^i = U$.
2) $\Gamma^{\gamma^i} \bigcap \Gamma^{\gamma^j} = \emptyset$.
3) $\sum_{\substack{i \in I \\ j \in \Gamma^k \ni \gamma^i \in \Gamma^k}} w_{i,j}$ *minimized over all M and P.*

The five following 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. The $RMLF$ considers medians selected from $S_n{}'$ and uses the d-distance.

Definition 2.3. *MLF' is the 2-median problem on $(K_H = (H, E(H, H)))$ with $w_{i,j} = d(v^i, v^j)$.*

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 $(K_H = (H, E(H, H)))$, which means the medians are located over the edge set of the complete graph on the fragments in H.

		SNP						
		1	2	3	4	5	6	7
Fragments	1	A	B	A	A	-	-	-
	2	B	A	B	B	-	-	-
	3	-	B	A	-	B	A	-
	4	-	-	B	B	B	B	-
	5	-	-	-	B	A	B	B
	6	-	-	-	A	B	B	A
	7	-	-	-	-	A	B	A
	8	-	-	-	-	B	A	A

Table 2: SNP Matrix

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.

Definition 2.4. *MLF is the 2-median problem on $(S_n, E(H, S_n))$ with $w_{i,j} = d(v^i, v^j)$.*

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.

Definition 2.5. *$DMLF'$ is the 2-median problem on $(K_H = (H, E(H, H)))$ with $w_{i,j} = l(v^i, v^j)$.*

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 with 11 flips required to make it conflict-free. 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.

Definition 2.6. *$DMLF$ is the 2-median problem on (S_n, $E(H, S_n)$) with $w_{i,j} = l(v^i, v^j)$.*

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 $\{ABAAABAA, BABBBABB\}$ as the medians with one required flip. This flip occurs when we relabel the 7^{th} fragment on the fifth SNP location from an A to a B.

Definition 2.7. *$RMLF$: 2-median problem on (K_{S_n}, $E(H, S_n{}')$) with $w_{i,j} = d(v^i, v^j)$.*

The $RMLF$ locates the two medians from the set $S_n{}'$, so we are restricting the medians to be complete fragments where each SNP is either A or B. This is the most biologically useful problem formulation because it locates two complete parent haplotypes without ambiguities. We often use the $RMLF$ as a comparison to the other problem solutions because we know it returns a completed haplotype while the others may not.

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 $CGMLF$.

3 MLF′ Algorithm

The algorithm presented here determines both the minimal number of flips and a $\{\gamma^1, \gamma^2\}$ 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 $(c = 1, n)$
    - if $d(a, c) < d(b, c)$ add $d(a, c)$ to flip count
    - else add $d(b, c)$ to flip count
    - end
  - set $d(\Gamma^a, \Gamma^b) =$ flip count
  - if $d(\Gamma^a, \Gamma^b) <$ min
    - set $\gamma^i = a$
    - set $\gamma^j = b$
    - set min $= d(\Gamma^a, \Gamma^b)$
  - end
- end.

This algorithm enumerates through each of the $\begin{pmatrix} m \\ 2 \end{pmatrix}$ 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 $v^i, v^j, ...$ We next show that it is optimal for each fragment to be assigned to its nearest median. Then we show that the algorithm produces the same $\{\gamma^1, \gamma^2\}$ we set out to find in our problem description.

Lemma 3.1. *Let v^1 and v^2 be vertices of the connected graph $g = (V, E)$. Then, the nearest vertex assignment (partition) minimizes the total distance of assigning V to $\{v^1, v^2\}$.*

Proof. Let $w_{i,j}$ be the distance between v^i and v^j. Let $\{V^1, V^2\}$ be the nearest vertex partition and $\{\hat{V}^1, \hat{V}^2\}$ be an arbitrary partition of V. We can say that

$$(V^1 \backslash \hat{V}^1) \bigcup (V^2 \backslash \hat{V}^2) = (\hat{V}^1 \backslash V^1) \bigcup (\hat{V}^2 \backslash V^2) \tag{1}$$

because

$$\begin{aligned}(V^1\backslash\hat{V}^1)\bigcup(V^2\backslash\hat{V}^2) &= (V^1\bigcap\hat{V}^2)\bigcup(V^2\bigcap\hat{V}^1) &(2)\\ &= (\hat{V}^2\bigcap V^1)\bigcup(\hat{V}^1\bigcap V^2) &(3)\\ &= (\hat{V}^2\backslash V^2)\bigcup(\hat{V}^1\backslash V^1). &(4)\end{aligned}$$

Then we can say

$$\sum_{k\in V^1\backslash\hat{V}^1} w_{1,k} + \sum_{k\in V^2\backslash\hat{V}^2} w_{2,k} \leq \sum_{k\in\hat{V}^1\backslash V^1} w_{1,k} + \sum_{k\in\hat{V}^2\backslash V^2} w_{2,k}. \quad (5)$$

Furthermore

$$\begin{aligned}\sum_{k\in V^1} w_{1,k} + \sum_{k\in V^2} w_{2,k} &= \sum_{k\in V^1\bigcap\hat{V}^1} w_{1,k} + \sum_{k\in V^1\backslash\hat{V}^1} w_{1,k} + \sum_{k\in V^2\bigcap\hat{V}^2} w_{2,k} + \sum_{k\in V^2\backslash\hat{V}^2} w_{2,k} &(6)\\ &\leq \sum_{k\in V^1\bigcap\hat{V}^1} w_{1,k} + \sum_{k\in\hat{V}^1\backslash V^1} w_{1,k} + \sum_{k\in V^2\bigcap\hat{V}^2} w_{2,k} + \sum_{k\in\hat{V}^2\backslash V^2} w_{2,k}. &(7)\end{aligned}$$

Hence, $\{V^1, V^2\}$ minimizes the distance of assigning V to $\{v^1, v^2\}$. □

Lemma 3.1 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 3.1. *Algorithm 1 generates an optimal pair of medians in polynomial time.*

Proof. Let Algorithm 1 terminate with $\{\gamma^1, \gamma^2\}$. Suppose that $\{v^1, v^2\}$ is a set of medians. From Lemma 3.1 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 $\{v^1, v^2\}$. 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. Since this distance is accomplished by Dijkstra's Algorithm ($O(m^2)$), the result is $O(m^5)$ on a connected non-complete graph. □

Corollary 3.1. *Algorithm 1 generates an optimal pair of medians in $O(m^3)$ for a complete graph.*

Proof. From Theorem 3.1, we can solve a 2-median problem on a connected graph in $O(m^5)$. Considering that Dijkstra's Algorithm is used to make a connected graph complete, it is not required for an existing complete graph. In this case, we do not need the m^2 iterations of Dijkstra's Algorithm, therefore Algorithm 1 is $O(m^3)$. □

We can extend these previous results to the general p-median problem where p is independent of the number of vertices. The following theorem states that we can solve an unconnected graph of m vertices in polynomial time where p is known.

Theorem 3.2. *The p-median problem on a non-complete, connected graph is polynomial with complexity* $O(m^{p+2})$.

Proof. Step 1: As was seen from the proof of Theorem **??**, to create each subset of fragments of size p (p-tuples), $\begin{pmatrix} m \\ p \end{pmatrix} = \frac{m!}{(m-p)!p!} = \frac{m(m-1)...(m-p+1)}{p!} = \frac{m^p - am^{p-1} + bm^{p-2} - ...}{p!}$ possible combinations exist on n fragments. For our purposes, the denominator and the values of a and b do not matter because they are integers and do not affect Big O analysis. Since m^p is the greatest power polynomial, we can disregard the others that follow. Therefore, $\begin{pmatrix} m \\ p \end{pmatrix}$ is solved in m^p.
Step 2: The inner loop uses Dijkstra's algorithm to find the shortest path between every pair of vertices. This completes the connected graph by assigning the distance between any two non-adjacent vertices to be the shortest path between them. Dijkstra's algorithm is known to be $O(m^2)$.
Step 3: To find the nearest median of each vertex, all n vertices must individually go through the current p-medians to find the minimum assignment. Hence this is done in pm.
Altogether, this yields a complexity of $pm^p(m^2)$, which is $O(m^{p+2})$. □

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.

3.1 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 γ^1 and γ^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 $\{\Gamma^{\gamma^1}, \Gamma^{\gamma^1}\}$ only denotes the set of fragments assigned to $\{\gamma^1, \gamma^2\}$; no SNPs have been changed to ensure that all fragments in Γ^{γ^1} are zero distance away from γ^1. $\hat{\Gamma}^{\gamma^1}$ is the set of altered fragments in Γ^{γ^1} after flipping has occurred. $\hat{\Gamma}^{\gamma^2}$ is defined similarly. Consider the following example:

$$\begin{gathered}\text{Given: } \gamma^1 = --\text{ABA} \\ \Gamma^{\gamma^1} = \{--\text{ABA, ABBBA, BBABB}\} \\ \text{flip count} \geq 2. \\ \hat{\Gamma}^{\gamma^1} = \{--\text{ABA, ABABA, BBABA}\}\end{gathered}$$

Notice that not all of the fragments $\Gamma^{\gamma^1} \underset{\sim}{\subseteq} \gamma^1$. However, each element in $\hat{\Gamma}^{\gamma^1} \underset{\sim}{\subseteq} \gamma^1$. Moreover, the flip count that Algorithm 1 returns indicates that this is the number of SNPs that have to be changed so that each element in $\hat{\Gamma}^{\gamma^1} \underset{\sim}{\subseteq} \gamma^1$ and each element in $\hat{\Gamma}^{\gamma^2} \underset{\sim}{\subseteq} \gamma^2$. 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 (flipped).

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}. Overlapping means that after we have our γ^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 γ^1 can be changed since all fragments in $\hat{\Gamma}^{\gamma^1} \underset{\sim}{\subseteq} \gamma^1$. 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 3.1. *Consider the following matrix,*

$$\begin{array}{cccc} A & B & - & - \\ B & B & A & - \\ - & A & B & A \\ - & - & A & B \end{array}$$

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* $\Gamma^{\gamma^2} = \{AB--, BBA-, --AA\}$.

Since the flip count = 0, $\{\Gamma^{\gamma^1}, \Gamma^{\gamma^1}\} = \{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^1}\}$. *In this case, overlapping completes the* $-$ *in the second SNP in* γ^2*, but does not complete the* $-$ *in the first SNP position in* γ^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* γ^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 Algorithm 1 returns.*

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 $\{\gamma^1, \gamma^2\} \underset{\sim}{\subseteq} RMLF$. Here is an example:

> Consider the matrix from Example 3.1. To find a set of optimal solutions, $RMLF$, we look through the set of 2^4 completely resolved fragments $\{A, B\}^4$. 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 $RMLF$ is {AABA, ABAA} since it only requires one flip (changing the first SNP in the second fragment to an A). Our $\{\gamma^1, \gamma^2\} = \{-\text{ABA}, --\text{AA}\}$ that Algorithm 1 returned is indeed $\underset{\sim}{\subseteq}$ of this optimal solution RMLF={AABA, ABAA}. Note that there exists other optimal solutions to $RMLF$, but all that is required is to show that our $\{\gamma^1, \gamma^2\} \underset{\sim}{\subseteq}$ 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 a subset of $RMLF$.

Example 3.2. *Case 1: One such case occurs when* $h^i \underset{\sim}{\subset} h^j$ *for some* $i, j \in H$*. Below is a matrix with fragments that have the relation* $\underset{\sim}{\subseteq}$ *whose* $\{\gamma^1, \gamma^2\}$ *is not a* $\underset{\sim}{\subseteq}$ *of* $RMLF$*.*

$$
\begin{array}{cccc}
B & - & A & A \\
A & A & A & - \\
- & B & A & A \\
A & - & A & - \\
* \; - & - & B & B \\
* \; - & - & A & A,
\end{array}
$$

*where 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* $\Gamma^{\gamma^2} = \{--AA,\ B-AA,\ AAA-,\ -BAA,\ A-A-\}$ *with flip count equal 0. Again, this implies that* $\{\Gamma^{\gamma^1}, \Gamma^{\gamma^1}\} = \{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^1}\}$*. However, to infer a* h^{*1} *and a* h^{*2} *using overlaps, it would take a minimum*

of 2 flips to make h^{*1} and h^{*2} (where one such solution would be $\{h^{*1}$,$h^{*2}\}$ = $\{AABA$, $AAAA\}$). However, suppose we let RMLF=$\{AAAA$, $BBAA\}$ 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 RMLF= $\{AAAA$, $BBAA\}$. However, this optimal solution RMLF= $\{AAAA$, $BBAA\}$, which only takes one flip, is not a $\underset{\sim}{\subset}$ of $\{\gamma^1, \gamma^2\}$=$\{--BB$,$--AA\}$ returned by Algorithm 1.

Case 2: The second case in which Algorithm 1 fails to give a $\{\gamma^1, \gamma^2\}$ that is $\underset{\sim}{\subset}$ of RMLF is the following:

*	A	A	-	-	-	-	-	-	-	
	B	B	B	B	B	A	-	-	-	-
	-	B	A	A	A	A	A	-	-	-
	-	-	B	B	B	B	B	-	-	-
	-	-	-	A	A	A	A	A	B	-
	-	-	-	-	-	B	B	B	B	B
*	-	-	-	-	-	-	-	-	A	A

Here, $\{\gamma^1, \gamma^2\}$=$\{AA-------$,$-------AA\}$, and the flip count Algorithm 1 returns is 0. Producing a h^{*1} and a h^{*2} by overlapping $\hat{\Gamma}^{\gamma^1}$ and $\hat{\Gamma}^{\gamma^2}$ requires 4 flips. However, suppose we let RMLF=$\{BBBBBBBBB$,$AAAAAAAAA\}$. It only takes 3 flips to construct RMLF from H, and $\{\gamma^1, \gamma^2\}$ is not *a subset of this RMLF.*

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 actual problem statement ($CGMLF$), a bipartite graph is required. What follows is an example that illustrates why MLF' does not produce a bipartite graph:

Example 3.3. *Consider the following simple SNP matrix:*

A	B	-	-
B	B	A	-
-	A	B	A
-	-	A	B

γ^1=$-ABA$, γ^2=$--AB$.
$\hat{\Gamma}^{\gamma^1} = \{-ABA\}$. $\hat{\Gamma}^{\gamma^2}$=$\{AB--$,$BBA-$,$--AB\}$.
flip count = 0.
The problem is that the partition $\hat{\Gamma}^{\gamma^2}$ is not bipartite because even though $AB--$ and

BBA− are zero distance away from their median −−AB, they are not zero distance away from each other (ie, 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 $CGMLF$.

4 DMLF′ Algorithm and DMLF Results

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. 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 $DMLF'$.

4.1 Optimal DMLF′ Algorithm

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

ALGORITHM 2

Step 0: Set 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 $(c = 1, n)$
    - if $l(a, c) < l(b, c)$ add $l(a, c)$ to flip count
    - else add $l(b, c)$ to flip count
    - end
  - set $l(\Gamma^a, \Gamma^b) =$ flip count
  - if $l(\Gamma^a, \Gamma^b) <$ min
    - set $\gamma^i = a$
    - set $\gamma^j = b$
    - set min $= l(\Gamma^a, \Gamma^b)$
  - end
- end.

Algorithm 2 finds the directed distance for each of the $2\times \begin{pmatrix} m \\ 2 \end{pmatrix}$ possible fragment pairs. There are two times the possible combinations because we have a distinction between (v^i, v^j) and (v^j, v^i) since their distances may not be the same. Algorithm 1, on the other hand, always considers (v^i, v^j) and (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.

4.2 Results

Due to the similarity of Algorithm 1 to Algorithm 2, the proofs of Lemma 3.1 and Theorem 3.1 (which deal with the optimality of the Algorithm 1) can easily be extended to apply to Algorithm 2. For this reason, the optimality argument for Algorithm 2 is omitted from this text and is left as an exercise to the reader.

Similar to the previous section on MLF', the following is a proof that $DMLF'$ is polyno-

mial.

Theorem 4.1. *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 $2\times \begin{pmatrix} m \\ 2 \end{pmatrix}=2\times \frac{(m^2-m)}{2}=(m^2-m)$ 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 $DMLF'$ is polynomial, we consider the complexity of $DMLF$. By Theorem 4.2, we see that $DMLF$ is solved in exponential time.

Theorem 4.2. $DMLF$ *is* $O(3^{3n})$

Proof. Recall from Theorem 4.1, that we can solve the 2-median porblem 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 $DMLF$ is solved in $O((3^n)^3)=O(3^{3n})$. □

Although $DMLF$ is not polynomial, we can test the solution in a polynomial number of steps. Generating the 3^n possibilities for fragments is what makes $DMLF$ non-polynomial; however, simply testing the 2-median solution is done in polynomial time. This implies that $DMLF$ is NP-complete.

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 −−−, in which case the distance from h^i to the median is 2. In other words, there is no benefit of having all − 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 γ^1 where there is no dash in the k^{th} SNP position of $\hat{h}^i$ (since the distance from $\hat{h}^i$ to γ^1 would not be zero) for all SNP positions k. Below is an example to illustrate this point:

> Suppose γ^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 γ^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 γ^1 is undefined, the following property about $DMLF'$ and $DMLF$ falls out nicely:

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

Another important observation about $DMLF'$ and $DMLF$ is how we build $\hat{h}$. Even though we know that $\hat{h}^i \underset{\sim}{\subset} \gamma^1$ $\forall h^i \in \Gamma^{\gamma^1}$, there are several options for what $\hat{h}^i$ could be. For example, given γ^1= AB−, and $h^i \in \Gamma^{\gamma^1}$ = 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.

5 CGMLF and DMLF

The $DMLF$ problem is, with the exception of the $RMLF$, 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 $\{s^r, s^t\}$ as the generated medians. In other words, $\{\gamma^1, \gamma^2\}$ are haplotypes the algorithm returns while $\{s^r, s^t\}$ are haplotypes that can be constructed from the partitions of $CGMLF$. Similarly, $\{\Gamma^{\gamma^1}, \Gamma^{\gamma^2}\}$ and $\{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^2}\}$ are used for the 2-median problems, while $\{H^1, H^2\}$ and $\{\hat{H}^1, \hat{H}^2\}$ 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 solving the $DMLF$ is equivalent to solving the $CGMLF$.

Lemma 5.1. *CG(H) is bipartite* $\Leftrightarrow \exists$ *a partition of* H*, say* $\{H^1, H^2\}$, $\ni$ *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

	SNP				
	1	2	3	4	5
1	A	B	A	-	-
Fragments 2	A	-	A	-	-
3	-	B	A	-	B
4	-	-	A	-	B
5	A	B	A	-	B
6	-	B	A	-	B
7	A	B	-	-	B
8	-	B	A	-	B

Table 3: An example of a feasible partition

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 5.1 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 a {A,-} column would conflict with the A SNPs and therefore is not biparitite, and likewise for A in a {B,-} column. 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 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 $\forall\ \hat{h}_k \in \hat{H}^i$, $\hat{h}_k \in \{A, B\}$ or $\hat{h}_k \in \{A, B\}$. To generate a parent haplotype, we define s^1 by:

$$s_k{}^1 = \begin{cases} A,\ if\ \ \hat{h_k} \in \{A, -\}\ \ for\ \ all\ \ \hat{h} \in \hat{H}^1 \\ B,\ if\ \ \hat{h_k} \in \{B, -\}\ \ for\ \ all\ \ \hat{h} \in \hat{H}^1 \\ -,\ if\ \ \hat{h_k} \in \{-\}\ \ for\ \ all\ \ \hat{h} \in \hat{H}^1, \end{cases}$$

for $k \in \{1...n\}$. We define $s_k{}^2$ 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 a $-$ 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 $DMLF$.

Theorem 5.1. $DMLF = CGMLF$

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 $\hat{h}^i \underset{\sim}{\subset} \gamma^1 \ \forall h^i \in \Gamma^{\gamma^1}$. By the definition of $\underset{\sim}{\subset}$, $h^i \underset{\sim}{\subset} \gamma^1$ iff ${h_k}^i = {\gamma_k}^1$ or ${h_k}^i = -$, $\forall k$. Since $\hat{h^i} \underset{\sim}{\subset} \gamma^1$, any $\hat{h}_k \in \hat{\Gamma}^{\gamma^1}$ is either in $\{A, -\}$, $\{B, -\}$, or just $\{-\}$. The same argument holds for all $\hat{h}^j \in \hat{\Gamma}^{\gamma^2}$. By Lemma 5.1, if $\exists$ a partition of H $(\{\hat{\Gamma}^{\gamma^1}, \hat{\Gamma}^{\gamma^2}\}) \ni$ every SNP partition set is either in $\{A, -\}$, $\{B, -\}$, or $\{-\}$, then CG(H) is bipartite. Thus, $DMLF$ is bipartite, and we say $DMLF \geq CGMLF$.

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 $d\{H, \hat{H^1} \bigcup \hat{H^2}\} = z(CGMLF)$. This notation comes from the definition of $CGMLF$; the distance from the set of fragments, H, to the two bipartite partitions $(\hat{H^1} \bigcup \hat{H^2})$ is precisely what is meant by the flip count of $CGMLF$. Since the partitions make a bipartite graph, the k^{th} SNP for every $\hat{h^i} \in \hat{H^1}$ is either in $\{A, -\}$ or $\{B, -\}$ $\forall k$. Following Rule X, let

$$ {s_k}^1 = \begin{cases} A, \ if \ \ \hat{h_k} \in \{A, -\} \ \ for \ \ all \ \ \hat{h} \in \hat{H^1} \\ B, \ if \ \ \hat{h_k} \in \{B, -\} \ \ for \ \ all \ \ \hat{h} \in \hat{H^1} \\ -, \ if \ \ \hat{h_k} \in \{-\} \ \ for \ \ all \ \ \hat{h} \in \hat{H^1}. \end{cases} $$

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 $d\{H, \hat{H^1} \bigcup \hat{H^2}\} = z(CGMLF)$, we know that

$$ z(CGMLF) = \sum_{\substack{j \\ h^i \in H}} d(h^i_j, \hat{h}^i_j) = \sum_{\substack{j \\ h^i \in H^1}} d(h^i_j, \hat{h}^i_j) \bigcup \sum_{\substack{j \\ h^i \in H^2}} d(h^i_j, \hat{h}^i_j). \tag{8} $$

Since the partitions are bipartite by definition of $CGMLF$, $\hat{H^1} \bigcap \hat{H^2} = 0$. We now write,

$$ z(CGMLF) = \sum_{\substack{j \\ h^i \in H}} d(h^i_j, \hat{h}^i_j) = \sum_{\substack{j \\ h^i \in H^1}} d(h^i_j, \hat{h}^i_j) + \sum_{\substack{j \\ h^i \in H^2}} d(h^i_j, \hat{h}^i_j). \tag{9} $$

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 $\{s^1, s^2\}$ as $\{\gamma^1, \gamma^2\}$ is not greater than $z(CGMLF)$, giving us a contradiction. Since we have chosen $\{s^1, s^2\}$ to be our $\{\gamma^1, \gamma^2\}$, $DMLF$ will assign each h^i to its nearest median and ultimately partition H into bipartite $\{\Gamma^{s^1}, \Gamma^{s^2}\}$. In other words, h^i takes some number of flips to become $\hat{h^i} \ni \hat{h^i} \underset{\sim}{\subset} s^r$. This point is reiterated with the following statement.

$$z(DMLF) = \sum_{h^i \in \Gamma^{s^1}} w_{i,s^1} + \sum_{h^i \in \Gamma^{s^2}} w_{i,s^2} = \tag{10}$$

$$\sum_{\substack{j \\ h^i \in \Gamma^{s^1}}} l(h^i_j, \hat{h}^i_j) + \sum_{\substack{j \\ h^i \in \Gamma^{s^1}}} d(\hat{h}^i_j, {s_j}^1) + \sum_{\substack{j \\ h^i \in \Gamma^{s^2}}} l(h^i_j, \hat{h}^i_j) + \sum_{\substack{j \\ h^i \in \Gamma^{s^2}}} d(\hat{h}^i_j, {s_j}^2). \tag{11}$$

However, by definition of $\hat{h^k}$, any $\hat{h^k}$ is conflict free of a parent haplotype, so $\sum_{\substack{j \\ h^i \in \Gamma^{s^1}}} d(\hat{h}^i_j, {s_j}^1)$ and $\sum_{\substack{j \\ h^i \in \Gamma^{s^2}}} d(\hat{h}^i_j, {s_j}^2)$ are 0. Therefore,

$$z(DMLF) = \sum_{\substack{j \\ h^i \in \Gamma^{s^1}}} l(h^i_j, \hat{h}^i_j) + \sum_{\substack{j \\ h^i \in \Gamma^{s^2}}} l(h^i_j, \hat{h}^i_j). \tag{12}$$

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 $\{\Gamma^{s^1}, \Gamma^{s^2}\}$ 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 $\hat{h^i} \underset{\sim}{\subset} s^r$, 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 12 becomes:

$$z(DMLF) = \sum_{\substack{j \\ h^i \in H^1}} d(h^i_j, \hat{h}^i_j) + \sum_{\substack{j \\ h^i \in H^2}} d(h^i_j, \hat{h}^i_j). \tag{13}$$

However, Equation 9, which gives $z(CGMLF)$ is identical to Equation 13, 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 $DMLF \geq CGMLF$, and from Part 2 that $z(DMLF) > z(CGMLF)$ is not true, we have proven that $DMLF = CGMLF$. □

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.

$CGMLF = DMLF$ is a significant result because we have a lot of results in place referring to $DMLF$. From the previous section, we shown that $DMLF$ is non-polynomial, which means we can equate $CGMLF$ to a NP-complete problem. Since we have built the relation between $CGMLF$ and $DMLF$, in the following section we attempt to construct relations between the remaining problem statements.

6 Chain of Inequalities

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 (z). We say problem statement I is $\leq$ 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 6.1. *Given a 2-median problem statement I and a 2-median problem statement J, where the distance measure 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 6.2. *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 $0 \leq c^1 \leq c^2$. In this case, c^1 is obviously the d-distance, while c^2 is the l-distance. □

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

Theorem 6.1. *$MLF \leq MLF' \leq DMLF'$ and $MLF \leq DMLF = CGMLF \leq DMLF'$.*

Proof. 1)$MLF \leq MLF' \leq DMLF'$.
$MLF \leq MLF'$ from Lemma 6.1 and $MLF' \leq DMLF'$ from Lemma 6.2. Thus, $MLF \leq MLF' \leq DMLF'$.
2)$MLF \leq DMLF = CGMLF \leq DMLF'$.
$MLF \leq DMLF$ from Lemma 6.2 and $DMLF \leq DMLF'$ from Lemma 6.1. In addition, from Theorem 5.1, $DMLF = CGMLF$. 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 done is the following:

> By Lemma 6.1, $DMLF$ should be less than or equal to MLF'.
> However, by Lemma 6.2, MLF' should be less than or equal to $DMLF$.

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 2-median problem in polynomial time.

In addition, we would like to fit $RMLF$ in this system of inequalities, but we have not yet explored this area. From Section 3, we know that $RMLF$ does not always intersect MLF'; however, no additional results are proved in this paper. We conjecture that $RMLF$ is equivalent to $DMLF$ and $CGMLF$ in every case with the exception of when every SNP in the SNP matrix is misread (which is not biologically probable). The following section restates our main results and proposes areas of further research.

7 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 l-distance over the complete set of fragments S_n. We can bound the $DMLF$, which is $O(3^{3n})$, from above by the $DMLF'$ in $O(m^3)$ time. We have further shown that the minimal flip count to solve the $MLF \leq MLF' \leq DMLF'$ and that the $MLF \leq DMLF = CGMLF \leq DMLF'$.

7.1 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 because they use different distance formulas and search for medians over different sets, completing this inequality is currently unavailable. 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.

Other future work would include investigating the type of sequencing errors that cause the MLF' and $DMLF'$ 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 MFR, MSR and the MLF problems, combining these error correcting techniques 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.

8 Acknowledgements

The research has been done under the Research Experience for Undergraduates (REU) program supported by the National Science Foundation. We would like to thank Dr. Allen Holder for his help as our advisor.

References

[1] British Computer Society, "An Algorithm for 2- Median Problem on Two Dimensional Meshes", *The Computer Journal*,Volume 44, Number 2,(2001), pp.1.

[2] James R. Evans and Edward Minieka, "Optimization Algorithms For Networks and Graphs." Second Edition, Marcel Dekker Inc. (1992), pp. 387.

[3] Harvey J. Greenberg, William E. Hart, Giuseppe Lancia, "Opportunities for Combinatorial Optimization in Computational Biology", preprint (2003).