Linear Time Probabilistic Algorithms for the Singular Haplotype Reconstruction Problem from SNP Fragments

Zhixiang Chen* Bin Fu* Robert Schweller* Boting Yang†
Zhiyu Zhao‡ Binhai Zhu§

Abstract

In this paper, we develop a probabilistic model to approach two scenarios in reality about the singular haplotype reconstruction problem - the incompleteness and inconsistency occurred in the DNA sequencing process to generate the input haplotype fragments and the common practice used to generate synthetic data in experimental algorithm studies. We design three algorithms in the model that can reconstruct the two unknown haplotypes from the given matrix of haplotype fragments with provable high probability and in time linear in the size of the input matrix. We also present experimental results that conform with the theoretical efficient performance of those algorithms. The software of our algorithms is available for public access and for real-time on-line demonstration.

1. Introduction

Most part of genomes between two humans are identical. The sites of genomes that make differences among human population are Single Nucleotide Polymorphisms (SNPs). The values of a set of SNPs on a particular chromosome copy define a *haplotype*. Haplotyping an individual involves determining a pair of haplotypes, one for each copy of a given chromosome according to some optimal objective functions.

In recent years, the haplotyping problem has been extensively studied (see, f.g., [1–8, 10–12]). There are several versions of the haplotyping problem. In this paper, we consider the singular haplotype reconstruction problem that asks to reconstruct two unknown haplotypes from the input matrix of fragments as accurately as possible. Like other versions of the problem, this has also been extensively studied (see, f.g., [2, 3, 10, 12]). Because both incompleteness and inconsistency are involved in the fragments, it is not surprising that various versions of the haplotyping problem are NP-hard or even hard to approximate (f.g., [2, 3, 10]), and many elegant and powerful methods such as those in [9] cannot be used to deal with incompleteness and inconsistency at the same time.

In this paper, we develop a probabilistic approach to overcome some of the difficulties caused by the incompleteness and inconsistency occurred in the input fragments. We design three algorithms in our probabilistic model that can reconstruct the two unknown haplotypes from the given matrix of haplotype fragments with provable high probability and in time linear in the size of the input matrix. We also present experimental results that confirm with the theoretical efficient performance of those algorithms. The software of our algorithms is available for public access and for real-time on-line demonstration.

*Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA. Emails: {chen, binfu, schwellerr}@cs.panam.edu.

†Department of Computer Science, University of Regina, Saskatchewan, S4S 0A2, Canada. Email: boting@cs.uregina.ca.

‡Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA. Email: zzha2@cs.uno.edu.

§Department of Computer Science, Montana State University, Bozeman, MT 59717. USA. Email: bhz@cs.montana.edu.

2. A Probabilistic Model

Assume that we have two haplotypes H_1, H_2, denoted as $H_1 = a_1a_2\cdots a_m$ and $H_2 = b_1b_2\cdots b_m$. Let $\Gamma = \{S_1, S_2, \ldots, S_n\}$ be a set of n fragments obtained from the DNA sequencing process with respect to the two haplotypes H_1 and H_2. In this case, each $S_i = c_1c_2\cdots c_m$ is either a fragment of H_1 or H_2. Because we lose the information concerning the DNA strand to which a fragment belongs, we do not know whether S_i is a fragment of H_1 or H_2. Suppose that S_i is a fragment of H_1. Because of reading errors or corruptions that may occur during the sequencing process, there is a small chance that either $c_j \neq$ - but $c_j \neq a_j$, or $c_j =$ -, for $1 \leq j \leq m$, where the symbol - denotes a hole or missing value. For the former, the information of the fragment S_i at the j-th SNP site is inconsistent, and we use α_1 to denote the rate of this type of inconsistency error. For the latter, the information of S_i at the j-th SNP site is incomplete, and we use α_2 to denote the rate of this type of incompleteness error. It is known (f.g., [2, 10, 12]) that α_1 and α_2 are in practice between 3% to 5%. Also, it is realistically reasonable to believe that the dissimilarity, denoted by β, between the two haplotypes H_1 and H_2 is big. Often, β is measured using the Hamming distance between H_1 and H_2 divided by the length m of H_1 and H_2, and is assumed to be large, say, $\beta \geq 0.2$. It is also often assumed that roughly half of the fragments in Γ are from each of the two haplotypes H_1 and H_2.

In the experimental studies of algorithmic solutions to the singular haplotype reconstruction problem, we often need to generate synthetic data to evaluate the performance and accuracy of a given algorithm. One common practice (f.g., [2, 10, 12]) is as follows: First, choose two haplotypes H_1 and H_2 such that the dissimilarity between H_1 and H_2 is at least β. Second, make n_i copies of H_i, $i = 1, 2$. Third, for each copy $H = a_1a_2\cdots a_m$ of H_i, for each $j = 1, 2, \ldots, m$, with probability α_1, flip a_j to a'_j so that they are inconsistent. Also, independently, a_j has probability α_2 to be a hole -. A synthetic data set is then generated by setting parameters m, $n_1, n_2, \beta, \alpha_1$ and α_2. Usually, n_1 is roughly the same as n_2, and $\beta \approx 0.2$, $\alpha_1 \in [0.01, 0.05]$, and $\alpha_2 \in [0.1, 0.3]$.

Motivated by the above reality of the sequencing process and the common practice in experimental algorithm studies, we will present a probabilistic model for the singular haplotype reconstruction problem. But first we need to introduce some necessary notations and definitions.

Let $\Sigma_1 = \{A, B\}$ and $\Sigma_2 = \{A, B, -\}$. For a set C, $|C|$ denotes the number of elements in C. For a fragment (or a sequence) $S = a_1a_2\cdots a_m \in \Sigma_2^m$, $S[i]$ denotes the character a_i, and $S[i, j]$ denotes the substring $a_i\cdots a_j$ for $1 \leq i \leq j \leq m$. $|S|$ denotes the length m of S. When no confusion arises, we alternatively use the terms fragment and sequence.

Let $G = g_1g_2\cdots g_m \in \Sigma_1^m$ be a fixed sequence of m characters. For any sequence $S = a_1\cdots a_m \in \Sigma_2^m$, S is called a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G)$ sequence if for each a_i, with probability at most α_1, a_i is not equal to g_i and $a_i \neq$ -; and with probability at most α_2, $a_i =$ -.

For a sequence S, define $holes(S)$ to be the number of holes in the sequence S. If A is a subset of $\{1, \cdots, m\}$ and S is a sequence of length m, $holes_A(S)$ is the number of $i \in A$ such that $S[i]$ is a hole.

For two sequences $S_1 = a_1\cdots a_m$ and $S_2 = b_1\cdots b_m$ of the same length m, for any $A \subseteq \{1, \cdots, m\}$, define

$$diff(S_1, S_2) = \frac{|\{i \in \{1, 2, \cdots, m\} | a_i \neq \text{ - and } b_i \neq \text{ - and } a_i \neq b_i\}|}{m}$$
$$diff_A(S_1, S_2) = \frac{|\{i \in A | a_i \neq \text{ - and } b_i \neq \text{ - and } a_i \neq b_i\}|}{|A|}.$$

For a set of sequences $\Gamma = \{S_1, S_2, \cdots, S_k\}$ of length m, define $vote(\Gamma)$ to be the sequence H of the same length m such that $H[i]$ is the most frequent character among $S_1[i], S_2[i], \cdots, S_k[i]$ for $i = 1, 2, \cdots, m$.

We often use an $n \times m$ matrix M to represent a list of n fragments from Σ_2^m and call M an SNP fragment matrix. For $1 \leq i \leq n$, let $M[i]$ represent the i-th row of M, i.e., $M[i]$ is a fragment in Σ_2^m.

We now define our probabilistic model:

The Probabilistic Singular Haplotype Reconstruction Problem: Let β, α_1 and α_2 be small positive constants. Let $G_1, G_2 \in \Sigma_1^m$ be two haplotypes with *diff*$(G_1, G_2) \geq \beta$. For any given $n \times m$ matrix M of SNP fragments such that n_i rows of M are $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences, $i = 1, 2$, $n_1 + n_2 = n$, reconstruct the two haplotypes G_1 and G_2, which are unknown to the users, from M as accurately as possible with high probability. We call β (resp., α_1, α_2) dissimilarity rate (resp., inconsistency error rate, incompleteness error rate).

3. Technical Lemmas

For probabilistic analysis we need the following two Chernoff bounds.

Lemma 1. *([9]) Let $X_1, \cdots, X_n$ be n independent random $0, 1$ variables, where X_i takes 1 with probability at most p. Let $X = \sum_{i=1}^n X_i$. Then for any $1 \geq \epsilon > 0$, $\Pr(X > pn + \epsilon n) < e^{-\frac{1}{3}n\epsilon^2}$.*

Lemma 2. *([9]) Let $X_1, \cdots, X_n$ be n independent random $0, 1$ variables, where X_i takes 1 with probability at least p. Let $X = \sum_{i=1}^n X_i$. Then for any $1 \geq \epsilon > 0$, $\Pr(X < pn - \epsilon n) < e^{-\frac{1}{2}n\epsilon^2}$.*

We shall prove several technical lemmas for algorithm analysis in the next three sections.

Lemma 3. *Let S be a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G)$ sequence. Then, for any $0 < \epsilon \leq 1$, with probability at most $2e^{-\frac{\epsilon^2 m}{3}}$, diff$(G_i, S) > \alpha_1 + \epsilon$ or holes$(S) > (\alpha_2 + \epsilon)m$.*

Proof: Let X_k, $k = 1 \cdots, m$, be random variables such that $X_k = 1$ if $S[k] \neq G_i[k]$ and $S[k] \neq -$, or 0 otherwise. By the definition of the $\mathcal{F}_{\alpha_1,\alpha_2}(m, G)$ sequences, X_k are independent and $Pr(X_k = 1) \leq \alpha_1$. So, by Lemma 1, with probability at most $e^{-\frac{\epsilon^2 m}{3}}$, $X_1 + \cdots + X_m > (\alpha_1 + \epsilon)m$ Thus, we have *diff*$(G, S) > \alpha_1 + \epsilon$ with probability at most $e^{-\frac{\epsilon^2 m}{3}}$. Similarly, with probability at most $e^{-\frac{\epsilon^2 m}{3}}$, $holes(S) > (\alpha_2 + \epsilon)m$. ∎

Lemma 4. *Assume that A is a fixed subset of $\{1, 2, \cdots, m\}$. Let S be a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G)$ sequence. Then, for any $0 < \epsilon \leq 1$, with probability at most $2e^{-\frac{\epsilon^2|A|}{3}}$, diff$_A(G_i, S) > \alpha_1 + \epsilon$ or* holes$_A(S) > (\alpha_2 + \epsilon)|A|$.

Proof: Let S' be the subsequence consisting of all the characters $S[i]$, $i \in A$, with the same order as in S. Similarly, let G' be the subsequence consisting of all the characters $G[i]$, $i \in A$, with the same order as in G. It is easy to see that *diff*$_A(S, G_i) =$ *diff*(S', G'). The lemma follows from a similar proof for Lemma 3. ∎

Lemma 5. *Let N_i be a set of n_i many $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences, $i = 1, 2$. Let β and ϵ be two positive constants such that $2\alpha_1 + 2\alpha_2 + 2\epsilon < 1$, and diff$(G_1, G_2) \geq \beta$. Then, with probability at most $2(n_1 + n_2)e^{-\frac{\epsilon^2\beta m}{3}}$, diff$(S_i, S_j) \leq \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$ for some $S_i \in N_i$ and some $S_j \in N_j$ with $i \neq j$.*

Proof: For each G_i, let A be the set of indexes $\{k \in \{1, 2, \cdots, m\} | G_i[k] \neq G_j[k]\}$, where $i \neq j$. Since *diff*$(G_i, G_j) \geq \beta$ and $|G_i| = |G_j| = m$, we have $|A| \geq \beta m$. For any $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequence S, by Lemma 4, with probability at most $2e^{-\frac{\epsilon^2|A|}{3}} \leq 2e^{-\frac{\epsilon^2\beta m}{3}}$, diff$_A(S, G_i) > \alpha_1 + \epsilon$ or holes$_A(S) > (\alpha_2 + \epsilon)|A|$. Hence, with probability at most $2n_i e^{-\frac{\epsilon^2\beta m}{3}}$, diff$_A(S, G_i) > \alpha_1 + \epsilon$ or holes$_A(S) > (\alpha_2 + \epsilon)|A|$, for some $S \in N_i$. Therefore, with probability at most $2(n_1 + n_2)e^{-\frac{\epsilon^2\beta m}{3}}$, we have diff$_A(S, G_i) > \alpha_1 + \epsilon$ or holes$_A(S) > (\alpha_2 + \epsilon)|A|$, for some $S \in N_i$, for some $i = 1$ or 2. In other words, with probability at least $1 - 2(n_1 + n_2)e^{-\frac{\epsilon^2\beta m}{3}}$, we have diff$_A(S, G_i) \leq \alpha_1 + \epsilon$ and holes$_A(S) \leq (\alpha_2 + \epsilon)|A|$, for all $S \in N_i$ and for $i = 1$ and 2.

For any $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequence S_i, $i = 1, 2$, if $\mathit{diff}_A(S_i, G_i) \leq \alpha_1 + \epsilon$ and $\mathit{holes}_A(S_i) \leq (\alpha_2 + \epsilon)|A|$, then $\mathit{diff}(S_1, S_2) \geq \mathit{diff}_A(S_1, S_2) \geq \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$. Thus, with probability at least $1 - 2(n_1 + n_2)e^{-\frac{\epsilon^2\beta m}{3}}$, we have $\mathit{diff}(S_1, S_2) \geq \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$, for every $S_1 \in N_1$ and every $S_2 \in N_2$. In words, with probability at most $2(n_1 + n_2)e^{-\frac{\epsilon^2\beta m}{3}}$, we have $\mathit{diff}(S_1, S_2) < \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$, for some $S_1 \in N_1$ and some $S_2 \in N_2$. ∎

Lemma 6. *Let α_1, α_2 and ϵ be three small positive constants that satisfy $0 < 2\alpha_1 + \alpha_2 - \epsilon < 1$. Assume that $N = \{S_1, \cdots, S_n\}$ is a set of $\mathcal{F}_{\alpha_1,\alpha_2}(m, G)$ sequences. Let $H = \mathit{vote}(N)$. Then, with probability at most $2m(e^{-\frac{\epsilon^2 n}{2}})$, $G \neq H$.*

Proof: Given any $1 \leq j \leq m$, for any $1 \leq i \leq n$, let X_i be random variables such that $X_i = 1$ if $S_i[j] \neq G[j]$ and $S_i[j] \neq -$, or 0 otherwise. By the definition of the $\mathcal{F}_{\alpha_1,\alpha_2}(m, G)$ sequences, X_i are independent and $Pr(X_i = 1) \leq \alpha_1$. So, by Lemma 2, with probability at most $e^{-\frac{\epsilon^2 n}{2}}$, $X_1 + \cdots + X_n < (\alpha_1 - \epsilon)n$. That is, with probability at most $e^{-\frac{\epsilon^2 n}{2}}$, there are fewer than $(\alpha_1 - \epsilon)n$ characters $S_i[j]$ such that $S_i[j] \neq G[j]$ and $S_i[j] \neq$ -. Similarly, with probability at most $e^{-\frac{\epsilon^2 n}{2}}$, there are fewer than $(\alpha_2 - \epsilon)n$ characters $S_i[j]$ such that $S_i[j] =$ -. Thus, with probability at most $2me^{-\frac{\epsilon^2 n}{2}}$, there are fewer than $(\alpha_1 + \alpha_2 - 2\epsilon)n$ characters $S_i[j]$ such that $S_i[j] \neq G[j]$ for some $1 \leq j \leq m$. This implies that, with probability at least $1 - 2me^{-\frac{\epsilon^2 n}{2}}$, there are more than $(1 - \alpha_1 - \alpha_2 + 2\epsilon)n$ characters $S_i[j]$ such that $S_i[j] = G[j]$ for any $1 \leq j \leq m$. Since $0 < 2\alpha_1 + \alpha_2 - \epsilon < 1$ by the assumption of the theorem, we have $(\alpha_1 + \epsilon)n < (1 - \alpha_1 - \alpha_2 + 2\epsilon)n$. This further implies that with probability at least $1 - 2me^{-\frac{\epsilon^2 n}{2}}$, $\mathit{vote}(N)[j] = G[j]$ for any $1 \leq j \leq m$, i.e., $\mathit{vote}(N) = G$. ∎

4. When the Inconsistency Error Parameter Is Known

Theorem 7. *Assume that $\alpha_1, \alpha_2, \beta$, and $\epsilon > 0$ are small positive constants that satisfy $4(\alpha_1 + \epsilon) < \beta$ and $0 < 2\alpha_1 + \alpha_2 - \epsilon < 1$. Let $G_1, G_2 \in \Sigma_1^m$ be the two unknown haplotypes such that $\mathit{diff}(G_1, G_2) \geq \beta$. Let M be any given $n \times m$ matrix of SNP fragments such that M has n_i fragments that are $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences, $i = 1, 2$, and $n_1 + n_2 = n$. There exists an $O(nm)$ time algorithm that can find two haplotypes H_1 and H_2 with probability at least $1 - 2ne^{-\frac{\epsilon^2 m}{3}} - 2me^{-\frac{\epsilon^2 n_1}{2}} - 2me^{-\frac{\epsilon^2 n_2}{2}}$ such that either $H_1 = G_1$ and $H_2 = G_2$, or $H_1 = G_2$ and $H_2 = G_1$.*

Proof: The algorithm, denoted as SHR-One, is described as follows.

Algorithm SHR-One

Input: M, an $n \times m$ matrix of SNP fragments.
Parameters α_1 and ϵ.
Output: Two haplotypes H_1 and H_2.
Set $\Gamma_1 = \Gamma_2 = \emptyset$.
Randomly select a fragment $r = M[j]$ for some $1 \leq j \leq n$.
For every fragment r' from M do
If ($\mathit{diff}(r, r') \leq 2(\alpha_1 + \epsilon)$) then put r' into Γ_1
Let $\Gamma_2 = M - \Gamma_1$.
Let $H_1 = \mathit{vote}(\Gamma_1)$ and $H_2 = \mathit{vote}(\Gamma_2)$.
return H_1 and H_2.

End of Algorithm

Claim 1. With probability at most $ne^{-\frac{\epsilon^2 m}{3}}$, we have either $\mathit{diff}(f, G_1) > \alpha_1 + \epsilon$ for some $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequence f in M, or $\mathit{diff}(g, G_1) > \alpha_1 + \epsilon$ for some $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_2)$ sequence g in M.

By Lemma 4, for any fragment $f = M[k]$ such that f is a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequence, with probability at most $e^{-\frac{\epsilon^2 m}{3}}$ we have $\mathit{diff}(f, G_1) > \alpha_1 + \epsilon$. Since there are n_1 many $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequences

in M, with probability at most $n_1 e^{-\frac{\epsilon^2 m}{3}}$, we have $diff(f, G_1) > \alpha_1 + \epsilon$ for some $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequence f in M. Similarly, with probability at most $n_2 e^{-\frac{\epsilon^2 m}{3}}$, we have $diff(g, G_2) > \alpha_1 + \epsilon$ for some $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_2)$ sequence g in M. Combining the above completes the proof for Claim 1.

Claim 2. Let M_i be the set of all the $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences in M, $i = 1, 2$. With probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, Γ_1 and Γ_2 is a permutation of M_1 and M_2.

By the assumption of the theorem, the fragment r of M is either a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequence or a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_2)$ sequence. We assume that the former is true. By Claim 1, with probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, we have $diff(f, G_1) \leq \alpha_1 + \epsilon$ for all $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequences f in M, and $diff(g, G_1) \leq \alpha_1 + \epsilon$ for all $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_2)$ sequences g in M. Hence, for any fragment r' in M, if r' is a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequence, then with probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, we have $diff(r, r') \leq diff(r, G_1) + diff(r', G_1) \leq 2(\alpha_1 + \epsilon)$. This means that, with probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, all $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_1)$ sequences in M will be included in Γ_1. Now, consider that r' is a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_2)$ sequence in M. Since $diff(G_1, G_2) \leq diff(G_1, r) + diff(r, G_2) \leq diff(G_1, r) + diff(r, r') + diff(r', G_2)$, we have $diff(r, r') \geq diff(G_1, G_2) - diff(G_1, r) - diff(G_2, r')$. By the given condition of $diff(G_1, G_2) \geq \beta$ and $4(\alpha_1 + \epsilon) < \beta$, with probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, we have $diff(r, r') \geq \beta - diff(G_1, r) - diff(G_2, r') \geq \beta - 2(\alpha_1 + \epsilon) > 2(\alpha_1 + \epsilon)$, i.e., r' will not be added to Γ_1. Therefore, with probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, $\Gamma_1 = M_1$ and $\Gamma_2 = M - \Gamma_1 = M_2$. Similarly, if r is a $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_2)$ sequence, with probability at least $1 - ne^{-\frac{\epsilon^2 m}{3}}$, $\Gamma_1 = M_2$ and $\Gamma_2 = M - \Gamma_1 = M_1$. This completes the proof of Claim 2.

Suppose that Γ_1 and Γ_2 is a permutation of M_1 and M_2. Say, without loss of generality, $\Gamma_1 = M_1$ and $\Gamma_2 = M_2$. By Lemma 6, with probability at most $2me^{-\frac{\epsilon^2 n_1}{2}} + 2me^{-\frac{\epsilon^2 n_2}{2}}$, $vote(\Gamma_1) \neq G_1$ or $vote(\Gamma_2) \neq G_2$. Hence, by Claim 2, with probability at most $2ne^{-\frac{\epsilon^2 m}{3}} + 2me^{-\frac{\epsilon^2 n_1}{2}} + 2me^{-\frac{\epsilon^2 n_2}{2}}$, $vote(\Gamma_1) \neq G_1$ or $vote(\Gamma_2) \neq G_2$.

Concerning the computational time of the algorithm, we need to compute the difference between the selected fragment r and each of the rest $n-1$ fragments in the matrix M. Finding the difference between r and r' takes $O(m)$ steps. So, the total computational time is $O(nm)$, which is linear in the size of the input matrix M. ∎

5. When Parameters Are Not Known

In this section, we consider the case that the parameters α_1, α_2 and β are unknown. However, we assume the existence of those parameters for the input matrix M of SNP fragments. We will show that in this case we can still reconstruct the two unknown haplotypes from M with high probability.

Theorem 8. *Assume that $\alpha_1, \alpha_2, \beta$, and $\epsilon > 0$ are small positive constants that satisfy $2\alpha_1 + 2\alpha_2 + 2\epsilon < 1$, $0 < 2\alpha_1 + \alpha_2 - \epsilon < 1$, and $\beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon) > 2(\alpha_1 + \epsilon)$. Let $G_1, G_2 \in \Sigma_1^m$ be the two unknown haplotypes such that $diff(G_1, G_2) \geq \beta$. Let M be any given $n \times m$ matrix of SNP fragments such that M has n_i fragments that are $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences, $i = 1, 2$, and $n_1 + n_2 = n$. Then, there exists an $O(umn)$ time algorithm that can find two haplotypes H_1 and H_2 with probability at least $1 - (1 - \gamma)^u - 4ne^{-\frac{\epsilon^2 \beta m}{3}} - 2me^{-\frac{\epsilon^2 n_1}{2}} - 2me^{-\frac{\epsilon^2 n_2}{2}}$ such that H_1, H_2 is a permutation of G_1, G_2, where $\gamma = \frac{n_1 n_2}{n(n-1)}$ and u is an integer parameter.*

Proof: The algorithm, denoted as SHR-Two, is described as follows.

Algorithm SHR-Two

Input: M, an $n \times m$ matrix M of SNP fragments.
u, a parameter to control the loop.

Output: two haplotypes H_1 and H_2.

Let $d_{\min} = \infty$ and $\mathcal{M} = \emptyset$.

For ($k = 1$ to u) do { //**the k-loop**

Let $M_1 = M_2 = \emptyset$ and $d_1 = d_2 = 0$.
Randomly select two fragments $r_1 = M[i_1], r_2 = M[i_2]$ from M
For every fragment r' from M do {
If ($\mathit{diff}(r_i, r') = min\{\mathit{diff}(r_1, r'), \mathit{diff}(r_2, r')\}$ for $i = 1$ or 2, then put r' into M_i.
}
Let $d_i = \max\{\mathit{diff}(r_i, r') | r' \in M_i\}$ for $i = 1, 2$.
Let $d = \max\{d_1, d_2\}$.
If ($d < d_{\min}$) then let $\mathcal{M} = \{M_1, M_2\}$ and $d_{min} = d$.
}
return $H_1 = \mathit{vote}(M_1)$ and $H_2 = \mathit{vote}(M_2)$.

End of Algorithm

Claim 3. With probability at most $(1-\gamma)^u$, r_1, r_2 is not a permutation of a $\mathcal{F}_{\alpha,\beta}(m, G_1)$ sequence and a $\mathcal{F}_{\alpha,\beta}(m, G_2)$ sequence in all of the k-loop iterations.

For randomly selected fragments r_1 and r_2, with probability γ, r_1, r_2 is a permutation of a $\mathcal{F}_{\alpha,\beta}(m, G_1)$ sequence and a $\mathcal{F}_{\alpha,\beta}(m, G_2)$ sequence in M. When the k-loop is repeated u times, with probability at most $(1-\gamma)^u$, r_1, r_2 is not a permutation of a $\mathcal{F}_{\alpha,\beta}(m, G_1)$ sequence and a $\mathcal{F}_{\alpha,\beta}(m, G_2)$ sequence in all of the u loop iterations. Thus, Claim 3 is true.

Let N_i be the set of the n_i fragments in M that are $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences, $i = 1, 2$.

Claim 4. With probability at most $4ne^{-\frac{\epsilon^2\beta m}{3}}$, $\mathit{diff}(G_i, S) > \alpha_1 + \epsilon$ or $\mathit{holes}(S) > (\alpha_2 + \epsilon)m$ for some S from N_i for some $i = 1$ or 2; or $\mathit{diff}(S_1, S_2) \leq \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$ for some $S_1 \in N_1$ and some $S_2 \in N_2$.

By Lemma 3, for every fragment S from N_i, with probability at most $2e^{-\frac{\epsilon^2 m}{3}}$, $\mathit{diff}(G_i, S) > \alpha_1 + \epsilon$ or S has more than $(\alpha_2 + \epsilon)m$ holes. Thus, with probability at most $2ne^{-\frac{\epsilon^2 m}{3}}$, $\mathit{diff}(G_i, S) > \alpha_1 + \epsilon$ or $\mathit{holes}(S) > (\alpha_2 + \epsilon)m$ for some S from N_i for some $i = 1$ or 2.

By Lemma 5, with probability at most $2ne^{-\frac{\epsilon^2\beta m}{3}}$, $\mathit{diff}(S_1, S_2) \leq \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$ for some $S_1 \in N_1$ and some $S_2 \in N_2$.

The above analysis completes the proof for Claim 4.

Claim 5. Let $H_1 = \mathit{vote}(M_1)$ and $H_2 = \mathit{vote}(M_2)$ be the two haplotypes returned by the algorithm. With probability at most $(1-\gamma)^u + 4ne^{-\frac{\epsilon^2\beta m}{3}}$, M_1, M_2 is not a permutation of N_1, N_2.

We assume that (1) $\mathit{diff}(S_1, S_2) > \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$ for every S_1 from N_1 and every S_2 from N_2; and (2) $\mathit{diff}(G_i, S) \leq \alpha_1 + \epsilon$ and $\mathit{holes}(S) \leq (\alpha_2 + \epsilon)m$ for all $S \in N_i$ for $i = 1, 2$. We consider possible choices of the two random fragments r_1 and r_2 in the following.

At any iteration of the k-loop, if $r_1 \in N_1$ and $r_2 \in N_2$, then by (2) we have $\mathit{diff}(r_1, r') \leq \mathit{diff}(r_1, G_1) + \mathit{diff}(r', G_1) \leq 2(\alpha_1 + \epsilon)$ for any $r' \in N_1$; and $\mathit{diff}(r_2, r') \leq \mathit{diff}(r_2, G_2) + \mathit{diff}(r', G_2) \leq 2(\alpha_1 + \epsilon)$ for any $r' \in N_2$. By (1) and the given condition of the theorem, we have, $\mathit{diff}(r_1, r') > \beta(1-2\alpha_1-2\alpha_2-2\epsilon) > 2(\alpha_1+\epsilon)$ for any $r' \in N_2$; and $\mathit{diff}(r_2, r') > \beta(1-2\alpha_1-2\alpha_2-2\epsilon) > 2(\alpha_1+\epsilon)$ for any $r' \in N_1$. This implies that at this loop iteration we have $M_1 = N_1, M_2 = N_2$ and $d \leq 2(\alpha_1 + \epsilon)$. Similarly, if at this iteration $r_1 \in N_2$ and $r_2 \in N_1$, then $M_1 = N_2, M_2 = N_1$ and $d \leq 2(\alpha_1 + \epsilon)$.

If $r_1, r_2 \in N_1$ at some iteration of the k-loop, then for any $r' \in N_2$, either $r' \in M_1$ or $r' \in M_2$. In either case, by (1) of our assumption and the given condition of the theorem, we have $d \geq \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon) > 2(\alpha_1 + \epsilon)$ at this iteration. Similarly, if $r_1, r_2 \in N_2$ at some iteration of the k-loop, then we also have $d > 2(\alpha_1 + \epsilon)$ at this iteration.

It follows from the above analysis that, under the assumption of (1) and (2), once we have $r_1 \in N_1$ and $r_2 \in N_2$ or $r_1 \in N_2$ and $r_2 \in N_1$ at some iteration of the k-loop, then M_1, M_2 is a permutation of N_1, N_2 at the end of this iteration. Furthermore, if M_1 and M_2 are replaced by M_1' and M_2' after this iteration, then M_1', M_2' must also be a permutation of N_1, N_2. By Claims 3 and 4, with probability at most $(1-\gamma)^u + 4ne^{-\frac{\epsilon^2\beta m}{3}}$, the assumption of (1) and (2) is not true, or $r_1 \in N_1$ and $r_2 \in N_2$ (or $r_1 \in N_2$ and $r_2 \in N_1$) is not true at all the iterations of the k-loop. Hence, with probability at most $(1-\gamma)^u + 4ne^{-\frac{\epsilon^2\beta m}{3}}$, the final list of M_1 and M_2 returned by the algorithm is not a permutation of N_1, N_2, so the claim is proved.

For M_1 and M_2 returned by the algorithm, we assume without loss of generality $M_i = N_i$, $i = 1, 2$. By Lemma 6 and the given condition of the theorem, with probability at most $2me^{-\frac{\epsilon^2 n_1}{2}} + 2me^{-\frac{\epsilon^2 n_2}{2}}$, we have $H_1 = vote(M_1) \neq G_1$ or $H_2 = vote(M_2) \neq G_2$. Thus, by Claim 5, with probability at most $(1-\gamma)^u + 4ne^{-\frac{\epsilon^2 \beta m}{3}} + 4me^{-\frac{\epsilon^2 n}{2}}$, we have $H_1 \neq G_1$ or $H_2 \neq G_2$.

It is easy to see that the time complexity of the algorithm is $O(umn)$, which is linear in the size of M. ∎

6. Tuning the Dissimilarity Measure

In this section, we consider a different dissimilarity measure in algorithm SHR-TWO to improve its ability to tolerate errors. We use the sum of the differences between r_i and every fragment $r' \in M_i$, $i = 1, 2$, to measure the dissimilarity of the fragments in M_i with r_i. The new algorithm SHR-Three is given in the following. We will present experimental results in Section 7 to show that algorithm SHR-Three is more reliable and robust in dealing with possible outliers in the data sets.

Algorithm SHR-Three

Input: M, an $n \times m$ matrix of SNP fragments.
u, a parameter to control the loop.

Output: two haplotypes H_1 and H_2.

Let $d_{\min} = \infty$ and $\mathcal{M} = \emptyset$.
For ($k = 1$ to u) do { //**the k-loop**
Let $M_1 = M_2 = \emptyset$ and $d_1 = d_2 = 0$.
Randomly select two fragments $r_1 = M[i_1], r_2 = M[i_2]$ from M
For every fragment r' from M do {
If ($diff(r_i, r') = min\{diff(r_1, r'), diff(r_2, r')\}$ for $i = 1$ or 2, then put r' into M_i.
}
Let $d_i = \sum_{r' \in M_i} diff(r_i, r')$ for $i = 1, 2$.
Let $d = \max\{d_1, d_2\}$.
If ($d < d_{\min}$) then let $\mathcal{M} = \{M_1, M_2\}$ and $d_{min} = d$.
}
return $H_1 = vote(M_1)$ and $H_2 = vote(M_2)$.

End of Algorithm

Theorem 9. *Assume that $\alpha_1, \alpha_2, \beta$, and $\epsilon > 0$ are small positive constants that satisfy $2\alpha_1 + 2\alpha_2 + 2\epsilon < 1$, $0 < 2\alpha_1 + \alpha_2 - \epsilon < 1$, and $\beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon) > 2(\alpha_1 + \epsilon)$. Let $G_1, G_2 \in \Sigma_1^m$ be the two unknown haplotypes such that $diff(G_1, G_2) \geq \beta$. Let M be any given $n \times m$ matrix of SNP fragments such that M has n_i fragments that are $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences, $i = 1, 2$, and $n_1 + n_2 = n$. Assume further that $\eta > \frac{2(\alpha_1+\epsilon)}{\beta(1-2\alpha_1-2\alpha_2-2\epsilon)}$ with $\eta = \frac{\min(n_1,n_2)}{2n}$. Then, there exists an $O(umn)$ time algorithm that can find two haplotypes H_1 and H_2 with probability at least $1-(1-\gamma)^u - 4ne^{-\frac{\epsilon^2\beta m}{3}} - 2me^{-\frac{\epsilon^2 n_1}{2}} - 2me^{-\frac{\epsilon^2 n_2}{2}}$ such that H_1, H_2 is a permutation of G_1, G_2, where $\gamma = \frac{n_1 n_2}{n(n-1)}$ and u is an integer parameter.*

Proof: Let N_i be the set of the n_i many $\mathcal{F}_{\alpha_1,\alpha_2}(m, G_i)$ sequences in M, for $i = 1, 2$.

We first notice that both Claims 3 and 4 in the proof of Theorem 8 hold here following the same analysis. However, we need to prove the following claim with different analysis:

Claim 6. Let $H_1 = vote(M_1)$ and $H_2 = vote(M_2)$ be the two haplotypes returned by the algorithm. With probability at most $(1-\gamma)^u + 4ne^{-\frac{\epsilon^2\beta m}{3}}$, M_1, M_2 is not a permutation of N_1, N_2.

We assume that (1) $diff(S_1, S_2) > \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$ for every S_1 from N_1 and every S_2 from N_2; and (2) $diff(G_i, S) \leq \alpha_1 + \epsilon$ and $holes(S) \leq (\alpha_2 + \epsilon)m$ for each S from N_i ($i = 1, 2$).

We shall consider the two cases:

Case 1. At some iteration of the k-loop, both r_1 and r_2 are selected from the same N_i for $i = 1$ or 2. For each $r' \in N_j$, $j \neq i$, by assumption (1) we have both $\mathit{diff}(r_i, r') > \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)$, $i = 1, 2$. Notice that r' must be either in M_1 or M_2. So, at least half of the fragments in N_j will be either in M_1 or M_2. Therefore, at this iteration, we have $d \geq \frac{1}{2} n_j \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon) \geq \eta n \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon) = \eta\beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon)n > 2(\alpha_1 + \epsilon)n$.

Case 2. At some iteration of the k-loop, r_1 and r_2 are selected from different N_i for $i = 1$ and 2. Without loss of generality, $r_i \in N_i$, $i = 1, 2$. For each $r' \in N_1$, by assumption (2) we have $\mathit{diff}(r_1, r') \leq 2(\alpha_1 + \epsilon)$; by assumption (1) and the given condition of the theorem we have $\mathit{diff}(r_2, r') > \beta(1 - 2\alpha_1 - 2\alpha_2 - 2\epsilon) > 2(\alpha_1 + \epsilon)$. Similarly, for each $r' \in N_2$, $\mathit{diff}(r_2, r') \leq 2(\alpha_1 + \epsilon)$, and $\mathit{diff}(r_1, r') > 2(\alpha_1 + \epsilon)$. Therefore, at this iteration, we have $M_1 = N_1$ and $M_2 = N_2$, and $d \leq 2(\alpha_1 + \epsilon)n_1 + 2(\alpha_1 + \epsilon)n_2 = 2(\alpha_1 + \epsilon)n$.

The two cases implies that under the assumption of (1) and (2), if at any iteration of the k-loop, we have $r_1 \in N_1$ and $r_2 \in N_2$, or $r_1 \in N_2$ and $r_2 \in N_1$, then the final list of the two sets M_1, M_2 is a permutation of N_1, N_2. Hence, Claim 6 follows from Claims 3 and 4 in the proof of Theorem 8, which are true here as we mentioned earlier.

Now, we assume that the final list of the two sets M_1, M_2 is a permutation of N_1, N_2. By Lemma 6, with probability at most $2m(e^{-\frac{\delta^2 n_1}{2}}) + 2m(e^{-\frac{\delta^2 n_2}{2}})$, $H_1 = \mathit{vote}(M_1)$, $H_2 = \mathit{vote}(M_2)$ is not a permutation of G_1, G_2. This, together with Claim 6, completes the probabilistic claim of the theorem.

It is easy to see that the time complexity of the algorithm is $O(umn)$, which is linear in the size of M. ∎

7. Experimental Results

We design a MATLAB program to test both the accuracy and the speed of algorithm SHR-Three. Due to the difficulty of getting real data from the public domain [1], our experiment data is created following the common practice in literature such as [1, 12]. A random matrix of SNP fragments is created as follows: (1) Haplotype 1 is generated at random with length m ($m \in \{50, 100, 150\}$). (2) Haplotype 2 is generated by copying all the bits from haplotype 1 and flipping each bit with probability β ($\beta \in \{0.1, 0.2, 0.3\}$). This simulates the dissimilarity rate β between two haplotypes. (3) Each haplotype is copied $\frac{n}{2}$ times so that the matrix has m columns and $n(n \in \{10, 20, 30\})$ rows. (4) Set each bit in the matrix to - with probability α_2 ($\alpha_2 \in \{0.1, 0.2, 0.3\}$). This simulates the incompleteness error rate α_2 in the matrix. (5) Flip each non-empty bit with probability $\alpha_1(\alpha_1 \in \{0.01, 0.02, ..., 0.1\})$. This is the simulation of the inconsistency error rate of α_1.

Tables 1 to 4 show the performance of algorithm SHR-Three with different parameter settings in accordance with those in [1]. The typical parameters used in [1] are $m = 100, n = 20, \beta = 0.2, \alpha_2 = 0.2$ and $0.01 \leq \alpha_1 \leq 0.05$. These are considered to be close to the real situations. In our tables, the results are the average time and the reconstruction rate of the 1000 executions of algorithm SHR-Three. A new random matrix is used for each execution. The reconstruction rate is defined as the ratio of the total number of correctly reconstructed bits to the total number of bits in two haplotypes. The computing environment is a PC machine with a typical configuration of 1.6GHz AMD Turion 64X2 CPUs and 1GB memory.

The software of our algorithms is available for public access and for real-time on-line demonstration at http://fpsa.cs.uno.edu/HapRec/HapRec.html. We thank Liqiang Wang for implementing the programs in Java and setting up this web site.

It should be pointed out that our work can be extended to reconstruct multiple haplotypes from a set of fragments. Our approach also opens the door to develop probabilistic methods for other variants of the haplotyping problem involving both inconsistency and incompleteness errors.

α_1 (%)	Time (ms)	Reconstruction Rate (%)
1	3.460	100.00
2	3.706	100.00
3	3.983	99.99
4	4.188	99.96
5	4.390	99.95
6	4.553	99.90
7	4.697	99.77
8	4.943	99.58
9	5.183	99.39
10	5.412	98.94

Table 1: Results for $m = 100, n = 20, \beta = 20\%$ and $\alpha_2 = 20\%$

	$n = 10$		$n = 30$	
α_1 (%)	Time (ms)	Reconstruction Rate (%)	Time (ms)	Reconstruction Rate (%)
1	2.444	99.91	4.744	100.00
2	2.568	99.78	5.046	100.00
3	2.674	99.58	5.261	100.00
4	2.774	99.36	5.605	99.99
5	2.851	99.01	6.045	100.00
6	2.925	98.60	6.302	99.97
7	3.028	98.03	6.567	99.96
8	3.121	97.54	6.870	99.85
9	3.213	96.81	7.307	99.70
10	3.314	95.85	7.635	99.56

Table 2: Results for $m = 100, \beta = 20\%, \alpha_2 = 20\%$

	$\beta = 10\%$		$\beta = 30\%$	
α_1 (%)	Time (ms)	Reconstruction Rate (%)	Time (ms)	Reconstruction Rate (%)
1	3.425	100.00	3.564	100.00
2	3.687	100.00	3.736	100.00
3	3.904	99.90	3.925	99.99
4	4.175	99.83	4.154	99.96
5	4.422	99.52	4.337	99.95
6	4.606	99.25	4.528	99.91
7	4.826	98.68	4.704	99.83
8	4.998	98.14	4.920	99.73
9	5.190	97.69	5.096	99.61
10	5.355	96.90	5.295	99.39

Table 3: Results for $m = 100, n = 20$ and $\alpha_2 = 20\%$

	$\alpha_2 = 10\%$		$\alpha_2 = 30\%$	
α_1 (%)	Time (ms)	Reconstruction Rate (%)	Time (ms)	Reconstruction Rate (%)
1	3.225	100.00	3.575	99.98
2	3.551	99.99	3.792	99.98
3	3.712	100.00	3.990	99.94
4	3.980	99.98	4.184	99.88
5	4.162	99.98	4.369	99.76
6	4.324	99.97	4.592	99.54
7	4.550	99.94	4.761	99.09
8	4.733	99.92	4.968	98.52
9	4.911	99.82	5.191	97.70
10	5.116	99.72	5.401	96.75

Table 4: Results for $m = 100, n = 20$ and $\beta = 20\%$

References

[1] M. S. Alessandro Panconesi. Fast Hare: A fast heuristic for single individual snp haplotype reconstruction. In *Algorithms in Bioinformatics, 4th International Workshop, WABI 2004, Lecture Notes in Computer Science 3240*, pages 266–277, 2004.

[2] V. Bafna, S. Istrail, G. Lancia, and R. Rizzi. Polynomial and apx-hard cases of the individual haplotyping problem. *Theoretical Computer Science*, 335:109–125, 2005.

[3] R. Cilibrasi, L. van Iersel, S. Kelk, and J. Tromp. On the complexity of several haplotyping problems. In *Algorithms in Bioinformatics, 5th International Workshop, WABI 2005, Lecture Notes in Computer Science*, volume 3692, pages 128–139, 2005.

[4] A. Clark. Inference of haplotypes from pcr-amplified samples of diploid populations. *Molecular Biology Evolution*, 7:111–122, 1990.

[5] D. Gusfield. A practical algorithm for optimal inference of haplotype from diploid populations. In *The Eighth International Conference on Intelligence Systems for Molecular Biology*, pages 183–189, 2000.

[6] D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *the Sixth Annual International Conference on Computational Biology*, pages 166–175, 2002.

[7] G. Lancia, M. C. Pinotti, and R. Rizzi. Haplotyping polulations by purs parsimongy: complexity and algorithms. *INFORMS Journal on computing*, 16:348–359, 2004.

[8] G. Lancia and R. Rizzi. A polynomial solution to a special case of the parsimongy haplotyping problem. *Operations Research letters*, 34:289–295, 2006.

[9] M. Li, B. Ma, and L. Wang. On the closest string and substring problems. *Journal of the ACM*, 49(2):157–171, 2002.

[10] R. Lippert, R. Schwartz, G. Lancia, and S. Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in bioinformatics*, 3:23–31, 2002.

[11] R. Rizzi, V. Bafna, S. Istrail, and G. Lancia. Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem. In *Algorithms in Bioinformatics: Second International Workshop, WABI 2002, Rome, Italy, September 17-21*, pages 29–43, 2002.

[12] R.-S. Wang, L.-Y. Wu, Z.-P. Li, and X.-S. Zhang. Haplotype reconstruction from SNP fragments by minimum error correction. *Bioinformatics*, 21(10):2456–2462, 2005.