Information-Theoretic Analysis of Haplotype Assembly

Hongbo Si, Haris Vikalo, and Sriram Vishwanath
Department of Electrical and Computer Engineering
The University of Texas at Austin
Email: hongbosi@utexas.edu, {hvikalo, sriram}@ece.utexas.edu

Abstract

This paper studies the haplotype assembly problem from an information-theoretic perspective. In the human genome, a haplotype is a sequence of nucleotide bases on a chromosome that differ from the bases in the corresponding positions on the other chromosome in a homologous pair. Haplotype sequences can conveniently be represented by binary strings, which enables us to transform the bioinformatics problem of haplotype assembly into an equivalent information-theoretic problem. Information about the order of bases in a genome is readily inferred using short reads provided by high-throughput DNA sequencing technologies. Performing haplotype assembly is challenging due to limited lengths of the reads and presence of sequencing errors. In this paper, the recovery of the target pair of haplotype sequences using short reads is transformed into an equivalent joint source-channel coding problem. Two binary messages, representing haplotypes and chromosome memberships of reads, are encoded and transmitted over a channel with erasures and errors, where the channel model reflects salient features of high-throughput sequencing. The focus of this paper is on determining the required number of reads for reliable haplotype reconstruction.

For the error-free reading case, erasure decoding is shown to be one of the optimal algorithms enabling reliable haplotype assembly. For the erroneous reading case, spectral partitioning is proven to be an efficient algorithm with order-wise optimal bounds.

The material in this paper was presented in part at the 2014 Information Theory Workshop, Hobart, Australia, Nov. 2014.

I. Introduction

Diploid organisms, including humans, have homologous pairs of chromosomes where one chromosome in a pair is inherited from mother and the other from father. The two chromosomes in a pair are structurally similar and basically carry the same type of information but are not identical. More specifically, chromosomes in a pair differ at a small fraction of positions (i.e., loci). Such variations are referred to as single nucleotide polymorphisms (SNPs); in humans, frequency of SNPs is approximately 1 base in 1000. A haplotype is the string of SNPs on a single chromosome in a homologous pair. Haplotype information is essential for understanding genetic causes of various diseases and for advancement of personalized medicine. However, direct analysis and identification of a haplotype is generally challenging, costly, and time and labor intensive. Alternatively, single individual haplotypes can be assembled from short reads provided by high-throughput sequencing systems. These systems rely on the so-called shotgun sequencing to oversample the genome and generate a redundant library of short reads. The reads are mapped to a reference and the individual genome is assembled following consensus of information provided by the reads. The length of each read (i.e., DNA fragment) in state-of-the-art sequencing systems is typically $100-1000$ base pairs [1]. Note that this length is comparable to the average distance between SNPs on chromosomes. Therefore, single reads rarely cover more than one variant site which is needed to enable haplotype assembly. Moreover, the origin of a read (i.e., to which chromosome in a pair the read belongs) is unknown and needs to be inferred [2]. Paired-end sequencing [3], also known as mate-paired sequencing [4], helps overcome these problems. This process generates pairs of short reads that are spaced along the target genome, where the spacing (so-called insert size) between the two reads in a pair is (approximately) known. The mate-pairs allow acquisition of the information about distant SNPs on the same chromosome, and thus help assemble the haplotype. Fig. 1 illustrates how paired-end reads may cover two or more variant sites along a homologous chromosome pair. The goal of haplotype assembly is to identify the chromosome from which fragments are sampled, and to reconstruct the haplotype sequences. When there are no sequencing errors, a fragment conflict graph framework [5] converts the original problem into partitioning of the set of reads into two subsets, each collecting the reads that belong to the same chromosome in a pair. For erroneous data, it poses haplotyping as an optimization problem of minimizing the number of

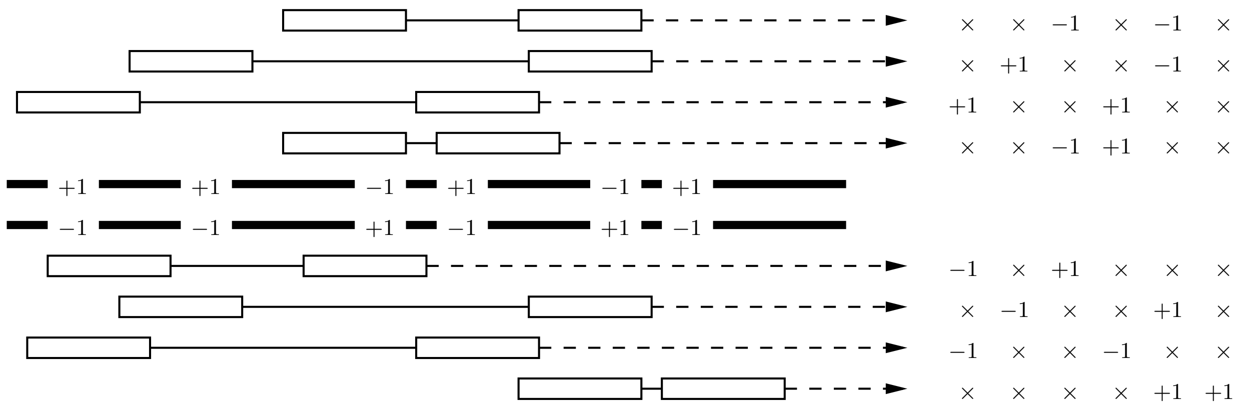

Fig. 1: Paired-end reads sampling two chromosomes in a homologous pair. Rectangles linked by the lines above and below the target chromosome pair represent paired-end reads, and their relative positions indicate their location along the chromosomes.

transformation steps needed to generate a bipartite graph [6]. This leads to various formulations of the haplotype assembly problem including minimum fragment removal (MFR), minimum SNP removal (MSR), and minimum error correction (MEC) [5]. The last one, MEC, has been the most widely used criterion for haplotype assembly, and is characterized by an inherent connection with the independent error model.

In this paper, we analyze the haplotype assembly problem from information-theoretic perspective, with emphasis on the necessary and sufficient conditions for reliable assembly. The contributions of the paper include: 1) an information-theoretic framework for haplotype assembly problem is proposed, i.e., two binary messages, representing haplotypes and chromosome memberships of reads, are encoded and transmitted over a channel with erasures and errors, where the channel model reflects salient features of high-throughput sequencing; 2) in the absence of reading errors, erasure decoding algorithm is shown to match the necessary condition on the number of reads required for recovery; 3) for the case where reading errors happen, spectral partitioning is proved to be an order-wise optimal algorithm enabling successful assembly; 4) algorithms proposed in the paper, although primarily meant to support theoretical results, also have practical significance.

The paper is organized as follows. Section II formalizes the haplotype assembly problem. In Section III, we present an information-theoretic view of haplotype assembly in the absence of sampling errors, while the erroneous case is discussed in Section IV. Simulation results and analyses are shown in Section V. Finally, Section VI concludes the paper.

II. Problem Formulation

As detailed in the introduction, a single nucleotide polymorphism (SNP) is a variation in a DNA sequence where two corresponding bases at a specific location on the homologous chromosomes differ from each other. Typically, diploid organisms have only two possible variants at a SNP site, i.e., their SNPs are typically biallelic. For the sake of convenience, we denote one of the two variants as $+1$ while the other one we denote as -1. With this notation, a haplotype sequence $\boldsymbol{h}$ comprising information about all SNP sites on one of the chromosomes in a homologous pair can be represented by a string with elements in $\{+1,-1\}$, while the haplotype associated with the other chromosome in the pair is its additive inverse $-\boldsymbol{h}$, where we denote

$$\boldsymbol{h} = (h_1, h_2, \dots, h_n),$$

and n is the length of haplotypes (i.e., the number of SNPs within each chromosome in a pair).

Each paired-end read acquired in a shotgun sequencing experiment contains partial information about either of these two haplotypes. Consider a set of discrete random variables c_i, where $i \in \{1,\dots,m\}$ and m denotes the number of reads. Let c_i identify the origin of read i, i.e., c_i carries information about the chromosome membership for read i. More precisely,

$$c_i = \begin{cases} +1, & \text{if read } i \text{ is sampled from } \boldsymbol{h}, \\ -1, & \text{if read } i \text{ is sampled from } -\boldsymbol{h}. \end{cases} \tag{1}$$

Due to the limitation of read lengths and relatively rare occurrence of SNPs, only a small fraction of variant sites is covered by a read. Formally, the information about a haplotype provided by a paired-end read $\boldsymbol{r}_i$ can be represented by a sequence that consists of symbols from the alphabet $\{+1,-1,\times\}$, where "$\times$" indicates lack of information about a variant site. Let us collect the relevant information provided by the reads in an $m \times n$ matrix $\boldsymbol{R}$ having rows corresponding to paired-end reads and columns corresponding to SNP sites. The ith row of $\boldsymbol{R}$ (i.e., read i) is denoted as $\boldsymbol{r}_i$, and the jth element of $\boldsymbol{r}_i$ is denoted as r_{ij}. Typically, since the length of a haplotype is much larger than the number of SNPs covered by a read, only few entries in each row are numerical (ignoring the occurrence of bursty variations).

Note that, in the absence of sampling noise, every observed element r_{ij} can be represented as the product of the jth SNP and the variable indicating membership of the ith read [7]. Formally, this can be written as

$$r_{ij} = c_i \cdot h_j. \tag{2}$$

From (2), matrix $\boldsymbol{R}$ could be interpreted as being obtained from a rank 1 matrix $\boldsymbol{S}$ whose row $\boldsymbol{s}_i$ is either $\boldsymbol{h}$ or $-\boldsymbol{h}$ based on the value of c_i, while most of its entries are erased in the reading process. In particular, we have

$$\boldsymbol{R} = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{S}), \text{ and } \boldsymbol{S} = \boldsymbol{c}^T \cdot \boldsymbol{h}, \tag{3}$$

where $\boldsymbol{\Omega}$ is the collection of all observed locations, and the projection $\mathcal{P}$ is defined by

$$\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{S})_{ij} = \begin{cases} s_{ij}, & \text{if } (i,j) \in \boldsymbol{\Omega}, \\ \times, & \text{if } (i,j) \notin \boldsymbol{\Omega}. \end{cases} \tag{4}$$

Hence, the task of haplotype assembly is to recover haplotype $\boldsymbol{h}$ and chromosome membership vector $\boldsymbol{c}$, or, equivalently, to find matrix $\boldsymbol{S}$ from matrix $\boldsymbol{R}$.

An example, illustrated by Fig. 1, corresponds to the scenario where 6 SNP sites are covered by 8 paired-end reads. The first 4 reads are assumed to be (shotgun) sequenced from chromosome 1 and thus the chromosome membership vector is $\boldsymbol{c} = (+1,+1,+1,+1,-1,-1,-1,-1)$. The true haplotype associated with chromosome 1 is assumed to be $\boldsymbol{h} = (+1,+1,-1,+1,-1,-1)$. In the absence of errors, the acquired SNP fragment matrix is given by

$$\boldsymbol{R} = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{c}^T \cdot \boldsymbol{h}) = \begin{bmatrix} \times & \times & -1 & \times & -1 & \times \\ \times & +1 & \times & \times & -1 & \times \\ +1 & \times & \times & +1 & \times & \times \\ \times & \times & -1 & +1 & \times & \times \\ -1 & \times & +1 & \times & \times & \times \\ \times & -1 & \times & \times & +1 & \times \\ -1 & \times & \times & -1 & \times & \times \\ \times & \times & \times & \times & +1 & +1 \end{bmatrix}. \tag{5}$$

III. Error-free Case

We first analyze haplotype assembly in the ideal scenario where the information provided by the sequencing reads is error-free. From a joint source-channel coding perspective, haplotype assembly aims to recover two sources being communicated through an erasure channel (see Fig. 2). The first source is haplotype information, $\boldsymbol{h}$, and the second source is the chromosome membership vector $\boldsymbol{c}$. Both of these vectors are assumed to originate from a uniform distribution, i.e., their entries have $1/2$ probability to take values from $\{+1,-1\}$. These two sources are encoded jointly using the function $f : \{+1,-1\}^n \times \{+1,-1\}^m \to \{+1,-1\}^{m\times n}$, and hence the encoded codeword $\boldsymbol{S} = f(\boldsymbol{h},\boldsymbol{c})$. In particular, each entry in $\boldsymbol{S}$ is given by $s_{ij} = c_i \cdot h_j$, which implies the encoder is a bijection.

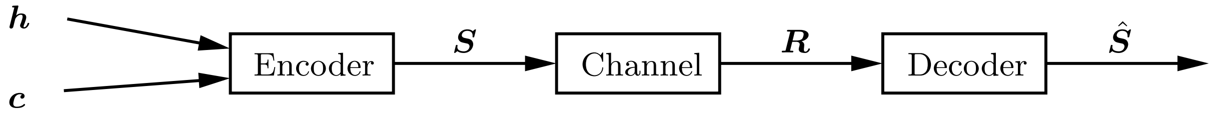

Fig. 2: Information theoretic view of the haplotype assembly problem.

After receiving the output from channel, $\boldsymbol{R}$, the decoder uses the decoding function to map its observations into an estimate of the message. Specifically, we consider the decoder (i.e., an algorithm for haplotype assembly) given by $g : \{+1,-1,\times\}^{m\times n} \to \{+1,-1\}^{m\times n}$, such that $\hat{\boldsymbol{S}} = g(\boldsymbol{R})$, where $\hat{\boldsymbol{S}}$ represents the estimate. Note that since the encoding function is a bijection, decoding $\boldsymbol{S}$ is equivalent to decoding both $\boldsymbol{h}$ and $\boldsymbol{c}$. We define the error probability of decoding as

$$P_{\mathrm{e}} \triangleq \Pr\{\hat{\boldsymbol{S}} \neq \boldsymbol{S}|\boldsymbol{R}\}. \tag{6}$$

As in the conventional information-theoretic analysis of a communication channel, we consider all possible choices of matrix $\boldsymbol{S}$ and denote the resulting ensemble by $\mathcal{S}$. Let m and n be sufficiently large so that there exists at least one decoding function g with small probability of error (similar type of analysis as in [8]). The channel model reflects particular reading technique. For the paired-end sequencing technique without sampling errors, let us consider the channel $W : \{+1,-1\}^{m\times n} \to \{+1,-1,\times\}^{m\times n}$ described as follows:

#1 Erasures happen independently across rows.

#2 In each row, only 2 entries remain and their positions are assumed to be uniformly placed. This can be easily extended to any number of (constant) entries within each row.

#3 Unerased entries are observed correctly.

In other words, for the sake of simplicity we assume that precisely 2 entries are observed in each row of $\boldsymbol{S}$, and that the observations are correct and independent across different rows. Under these assumptions, the number of numerical entries in each column of $\boldsymbol{R}$ approximately obeys Poisson distribution. Moreover, the expected length of the inserts between 2 sampled entries within a row is given by $(n-2)/3$. In practice, the inserts' length is limited and cannot be made arbitrarily large – a constraint that we relax in our analysis by making the assumption #2 above. Note that while we need this assumption to facilitate the theoretical analysis, the algorithms for haplotype assembly discussed in the paper do not make any assumptions on the insert lengths.

Based on this model, we derive the necessary and sufficient conditions on the number of error-free reads needed for haplotype assembly.

Theorem 1. *Given the SNP fragment matrix* $\boldsymbol{R}$ *with* 2 *reliable observations at arbitrary positions in each row, the original haplotype matrix* $\boldsymbol{S}$ *can be reconstructed if and only if the number of reads satisfies*

$$m = \Theta(n \ln n),$$

where n *is the length of the target haplotype, and the scaling factor can be chosen as* $1/2$.

To show the proofs of necessary and sufficient conditions, we provide a random graph interpretation of the haplotype assembly problem. More precisely, every SNP site is described by a node in a plane, and each paired-end read covering two SNP sites is described by an edge connecting two nodes. Then, the paired-end reading process can be depicted by randomly choosing two nodes and adding an edge between them. Based on this interpretation, a haplotype can be recovered with high reliability if and only if the graph resulting from the paired-end reading process is connected with high probability. The intuition of this observation is evident: if a node is not connected by any edge (i.e., if we have an unconnected graph), the corresponding SNP cannot be incorporated into the haplotype sequence and the assembly fails; on the other hand, if the graph is connected, there exist a path between any two nodes and thus a message passing algorithm can be designed to reconstruct the haplotype. For example, the following simple and effective algorithm, called “erasure decoding”, can be utilized for haplotype recovery:

1) Choose the “seed” s as an arbitrary non-erased entry in the first row, i.e., $s = r_{1j}$, where j is randomly chosen such that $r_{1j} \neq \times$. Set the chromosome membership variable of the first row to $c_1 = +1$ (the initial choice can be arbitrary).
2) Find all other rows with position j not erased, i.e., form a set
$$\mathcal{A} = \{k | r_{kj} \neq \times,\ k \neq 1\}. \tag{7}$$
3) Set the chromosome membership variables of the rows with indices in $\mathcal{A}$ to
$$c_k = \begin{cases} +1, & \text{if } r_{kj} = r_{1j}, \\ -1, & \text{otherwise}, \end{cases} \tag{8}$$
for every $k \in \mathcal{A}$.
4) Decode SNPs in the first row by evaluating
$$r_{1l} = c_k \cdot r_{kl}, \tag{9}$$
for every $k \in \mathcal{A}$ and $r_{kl} \neq \times$.
5) Delete all rows with indices in $\mathcal{A}$.
6) Arbitrarily choose another non-erased entry in the first row as the new seed $s = r_{1j}$ which has not been chosen as a seed in any of the previous steps. Repeat Step 2) to 6) until no row could be further erased.
7) If the first row is the only remaining one and its entries are all decoded, declare $\boldsymbol{h} = \boldsymbol{r}_1$; otherwise, declare a failure.

The key idea behind the erasure decoding algorithm is that a node can be decoded (i.e., its position within the haplotype sequence decided) using the previously decoded node on the other end of an edge. Note that in the error-free case, having a connected graph guarantees that the erasure decoding algorithm will successfully assemble the haplotype because all the entries in the first row can be decoded (otherwise at least one node is not connected). In this sense, erasure decoding can be considered as one of the optimal algorithms for haplotype assembly. Proceeding with the random graph interpretation, haplotype assembly under the error-free assumption is effectively converted to a random graph’s connectivity problem, where the latter is well-studied and the conditions on the number of edges needed to ensure connectivity are known. The proofs, i.e., the derivations of the lower and upper bounds, are presented in Appendix A.

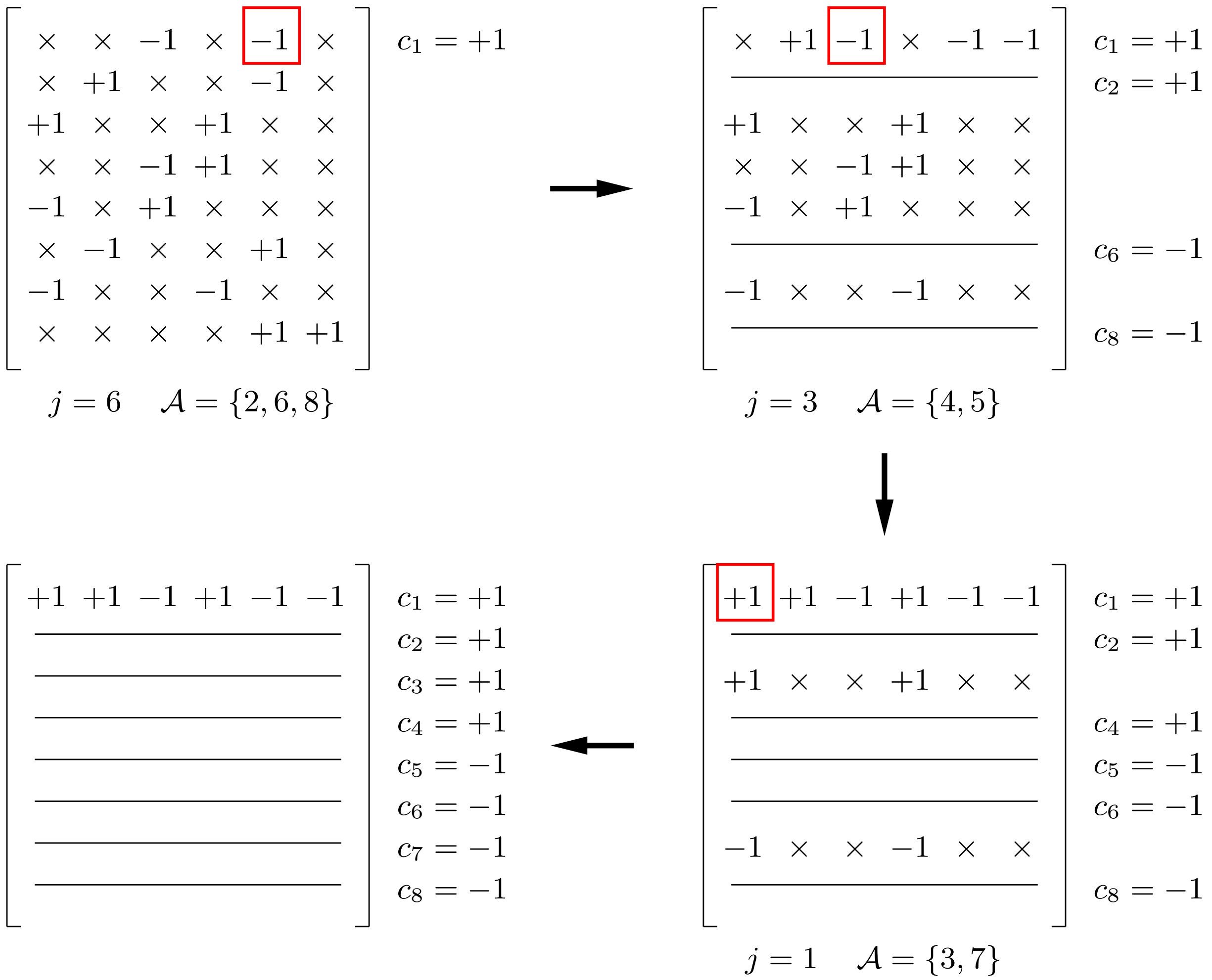

Fig. 3: Erasure decoding of the example illustrated in Fig. 1. In every round (steps 2 to 6), the seed is marked by a rectangle, with its column index given by j. Rows that share the same positions as the seed are collected in the set $\mathcal{A}$. A straight line crossing an entire row of the matrix represents a deletion.

Fig. 3 shows the details of the decoding procedure for the example illustrated in Fig. 1, where the read matrix is given by (5).

IV. Erroneous Case

When determining a component of the haplotype sequence at a particular position, we essentially need to perform a hypothesis test and decide between possible symbols in the corresponding column of the SNP fragment matrix. If sequencing errors are present, some of the entries in $\boldsymbol{R}$ are erroneously flipped. For the purpose of the following discussion, we assume such errors are independent and identically distributed (i.i.d.). More precisely, the errors are modeled as having originated by passing messages (i.e., the numerical entries in $\boldsymbol{R}$) through a collection of independent symmetric channels characterized by the parameter p, the probability of flipping the sign of the numerical entries of $\boldsymbol{R}$. Denoting the noise as matrix $\boldsymbol{N}$ with entries n_{ij} that are i.i.d., we can write

$$\boldsymbol{R} = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{S} \oplus \boldsymbol{N}). \tag{10}$$

Hence, the model describing the erroneous case is as same as the one for the error-free case except for an additional noise term capturing the effects of "channel" (i.e., the effects of sequencing and data processing steps that precede haplotype assembly). The equivalent channel model $W : \{+1,-1\}^{m\times n} \to \{+1,-1,\times\}^{m\times n}$ considered in this section is described as follows:

#1 Erasures happen independently across rows.

#2 In each row, only 2 entries remain and their positions are uniformly random.

#3 The remaining entries are read incorrectly with probability p and the errors are independent.

We would like to reconstruct $\boldsymbol{S}$ from $\boldsymbol{R}$ with high probability. However, if no more than two numerical entries are observed in a row, solving this problem is not always feasible. Assume, for instance, that the observed numerical entries in $\boldsymbol{r}_i$ are $(+1,+1)$, and that only one sequencing error happened (i.e., either one of $\boldsymbol{r}_i$'s numerical entries is erroneous). Then, there is no hope to discover whether the true numerical entries in $\boldsymbol{s}_i$ are $(-1,+1)$ or $(+1,-1)$. For this reason, in the erroneous case we aim to recover (with high probability) only the row space, i.e., find the haplotype $\boldsymbol{h}$ from matrix $\boldsymbol{R}$. Let us denote the haplotype estimate found by an assembly algorithm by $\hat{\boldsymbol{h}}$. We define the probability of error as

$$P_{\mathrm{e}} = \Pr\{\hat{\boldsymbol{h}} \neq \boldsymbol{h} | \boldsymbol{R}\},$$

and use it to characterize the accuracy of assembly. We would like to make this probability arbitrarily small on average (averaged over all possible $\boldsymbol{h}$).

Based on the previously described model of the haplotype assembly problem, we next state the necessary and sufficient conditions on the number of reads required for assembly.

Theorem 2. *Given the SNP fragment matrix* $\boldsymbol{R}$ *with* 2 *unreliable observations at arbitrary positions in each row, the original haplotype vector* $\boldsymbol{h}$ *can be reconstructed if and only if the number of reads satisfies*

$$m = \Theta(n \ln n),$$

where n denotes the length of the target haplotype.

The preceding theorem shows that although the observations are unreliable due to the sampling noise, the number of reads required for assembly still scales the same way as in the noise-free case, with the scaling factor now related to the sampling error probability. The proof for necessary condition follows the same line of arguments as in the error-free case; we provide a detailed proof for sufficient conditions as follow.

Recall that for the scenario where $\boldsymbol{R}$ is error-free, in Section III we proposed a random graph interpretation of haplotype assembly. Most state-of-the-art algorithms traces the same interpretation for the erroneous case and consider optimization formulations focusing on different objective criteria [5] to convert the erroneous graph into an error-free one. Formulations of the haplotype assembly problem include minimum fragment removal (MFR), minimum SNP removal (MSR), and minimum error correction (MEC). MFR [5] formulation aims to identify the smallest number of fragments (i.e., reads) whose removal renders the graph representing the assembly problem bipartite. Since the resulting graph is conflict-free, algorithms for error-free case could be readily applied to assemble the haplotypes. However, solving the MFR formulation of the assembly problem is challenging since the resulting optimization is generally non-convex. MSR [5] is an alternative formulation focused on identifying the smallest possible number of SNP sites such that the graph representing remaining SNPs could be partitioned in two subgraphs corresponding to the haplotypes. In graph-theoretic terms, MSR aims to find the maximum independent set of the original graph. MEC [6] formulation seeks the smallest number of entries in matrix $\boldsymbol{R}$ whose flipping ensures that the rows in $\boldsymbol{R}$ are consistent with having originated from two complementary haplotypes. In this formulation, the problem becomes the one of error-correction of binary data corrupted by i.i.d. noise. MEC is the most widely used formulation of the haplotype assembly problem, and a large number of algorithms have been developed for solving it (perhaps the most widely used one is HapCUT [9]).

However, in the presence of sampling errors, graphical interpretations of haplotype assembly do not provide obvious arguments for the fundamental requirements on the number of reads. To this end, we present an alternative low-rank matrix interpretation of the haplotype assembly problem. Intuitively, we aim to partition SNPs into two sets, each corresponding to one of the two haplotypes in a homologous pair. By regarding the adjacency matrix of the original graphical representation of the problem as a perturbation of a planted model (which is inherently a low rank matrix), we claim that the partition is perfect as long as the parameters of the model are chosen appropriately. In what follows, we first describe the "spectral partitioning" algorithm that relies on the singular value decomposition (SVD) technique to obtain a weaker conclusion that the fraction of partition errors vanishes as n increases, and then propose a modified algorithm for near-perfect haplotype assembly. The steps of the spectral partitioning algorithms are as follows:

1) Construct an adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$ based on the observation matrix $\boldsymbol{R}$, such that for every $(u,v) \in \{1,\ldots,n\} \times \{1,\ldots,n\}$ with $u > v$,

$$a_{uv} = \begin{cases} 1, & \text{if } \sum_{i=1}^{m} \mathbf{1}_{\{r_{iu}\neq\times, r_{iv}\neq\times, r_{iu}=r_{iv}\}} > \sum_{i=1}^{m} \mathbf{1}_{\{r_{iu}\neq\times, r_{iv}\neq\times, r_{iu}\neq r_{iv}\}}, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

   Then, let $a_{uv} = a_{vu}$ for any $u > v$ to guarantee symmetry, and let $a_{uu} = 0$ to enforce zeros on the diagonal of $\boldsymbol{A}$.

2) Find the singular value decomposition (SVD) of $\boldsymbol{A}$, i.e., $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}$ such that $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times n}$ are unitary matrices and $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ is diagonal.

3) Identify the eigenvector $\boldsymbol{v}_2(\boldsymbol{A})$ corresponding to the second largest eigenvalue of $\boldsymbol{A}$ and construct sets

$$\mathcal{C}_1 = \{j : v_{2j} < 0\}, \quad \mathcal{C}_2 = \{j : v_{2j} \geq 0\}.$$

   The haplotype is then recovered as

$$h_j = \begin{cases} +1, & \text{if } j \in \mathcal{C}_1, \\ -1, & \text{if } j \in \mathcal{C}_2. \end{cases}$$

Remark 3. *As evident from* (11)*, elements of* $\boldsymbol{A}$ *are evaluated by examining each SNP position and performing the majority voting operation over read components that cover it. This procedure is equivalent to the MAP hypothesis testing that assumes uniform SNP prior distribution. If the* *distribution of SNPs is not uniform, or if error distributions are not identical across SNP sites, one should rely on weighted majority voting instead.*

We analyze the performance of spectral partitioning by showing its relation to the classical partitioning problem on a planted model. This approach originates from the perturbation theory for eigenvectors and follows steps similar to those in [10], but additionally exploits structural features of the haplotype assembly problem to arrive at bounds that are much tighter than those provided in the general case.

A. Planted Model

Consider the planted model, i.e., matrix $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ defined as

$$\boldsymbol{B} = \begin{bmatrix} [\alpha]_{n_1 \times n_1} & [\beta]_{n_1 \times n_2} \\ [\beta]_{n_2 \times n_1} & [\alpha]_{n_2 \times n_2} \end{bmatrix},$$

where $\alpha > \beta > 0$, $n_1 + n_2 = n$, and $[\alpha]_{n_1 \times n_1}$ denotes an $n_1 \times n_1$ sub-matrix with all entries equal to α. Clearly, such a matrix $\boldsymbol{B}$ is low-rank. More precisely, if we perform the SVD on $\boldsymbol{B}$, it becomes evident that the rank of $\boldsymbol{B}$ is 2 and that its first two singular values and the corresponding singular vectors are given by

$$\lambda_1(\boldsymbol{B}) = n_1 \beta \mu_1 + n_2 \alpha \;, \tag{12}$$

$$\lambda_2(\boldsymbol{B}) = n_1 \beta \mu_2 + n_2 \alpha \;, \tag{13}$$

$$\boldsymbol{v}_1(\boldsymbol{B}) = \left(\left[\frac{\mu_1}{\sqrt{n_1 \mu_1^2 + n_2}} \right]_{1 \times n_1}, \left[\frac{1}{\sqrt{n_1 \mu_1^2 + n_2}} \right]_{1 \times n_2} \right), \tag{14}$$

$$\boldsymbol{v}_2(\boldsymbol{B}) = \left(\left[\frac{\mu_2}{\sqrt{n_1 \mu_2^2 + n_2}} \right]_{1 \times n_1}, \left[\frac{1}{\sqrt{n_1 \mu_2^2 + n_2}} \right]_{1 \times n_2} \right), \tag{15}$$

where

$$\mu_1 = \frac{(n_1 - n_2)\alpha + \sqrt{(n_1 - n_2)^2 \alpha^2 + 4 n_1 n_2 \beta^2}}{2 n_1 \beta} \;, \tag{16}$$

$$\mu_2 = \frac{(n_1 - n_2)\alpha - \sqrt{(n_1 - n_2)^2 \alpha^2 + 4 n_1 n_2 \beta^2}}{2 n_1 \beta} \;. \tag{17}$$

Note that since $\mu_1 > 0$ and $\mu_2 < 0$ for any n_1 and n_2, it holds that $\lambda_1(\boldsymbol{B}) > \lambda_2(\boldsymbol{B})$. Moreover, since $\mu_2 < 0$, the first n_1 entries in $\boldsymbol{v}_2(\boldsymbol{B})$ have opposite signs from those of the last n_2 entries.

Therefore, if we partition the indices into two sets with respect to their signs in $\boldsymbol{v}_2(\boldsymbol{B})$, the result naturally provides a classification corresponding to different blocks of matrix $\boldsymbol{B}$.

B. Adjacency Matrix Generated from the Planted Model

As discussed above, eigenvector corresponding to the second largest eigenvalue of the planted model $\boldsymbol{B}$ enables partitioning, i.e., helps distinguish between different block indices. The next step is to relate the planted model $\boldsymbol{B}$ to the adjacency matrix $\boldsymbol{A}$ constructed according to (11). Note that the entries in the upper-triangular part of $\boldsymbol{A}$ are random and independent. In fact, the distribution of each entry in $\boldsymbol{A}$ is Bernoulli with parameters which only depend on whether the corresponding SNP sites belong to the same block or not (i.e., two parameters are sufficient to characterize the distribution of $\boldsymbol{A}$). $\boldsymbol{A}$ and $\boldsymbol{B}$ are related through a series of permutations of rows and columns (note that permutations do not impact the eigenvectors). In particular, for any $(u,v) \in \{1,\ldots,n\} \times \{1,\ldots,n\}$ with $u > v$, we define

$$\begin{aligned} \Pr\{a_{uv} = 1\} &= \pi(b_{uv}), \\ \Pr\{a_{uv} = 0\} &= 1 - \pi(b_{uv}), \end{aligned}$$

where π is the permutation of rows and columns. Let α denote the probability that two SNP sites from the same cluster are inferred correctly in the majority voting step, while β denotes the probability that two SNP sites from different clusters are inferred incorrectly. Clearly, α and β are closely related to the accuracy and redundancy in the sequencing data – more precisely, the parameters n, m, and p. In our case of unreliable paired-end sequencing, the probabilities α and β are given by

$$\begin{aligned} \alpha &\triangleq \Pr\{\text{majority voting claims } a_{uv} = 1 | h_u = h_v\} \\ &= \sum_{i=1}^{m} \Pr\{\text{majority voting claims } a_{uv} = 1, \ i \text{ reads cover SNP sites } u \text{ and } v | h_u = h_v\} \\ &= \sum_{i=1}^{m} \left\{ \binom{m}{i} \left[\frac{2}{n(n-1)}\right]^i \left[1 - \frac{2}{n(n-1)}\right]^{m-i} \sum_{l=\lfloor i/2 \rfloor + 1}^{i} \binom{i}{l} [(1-p)^2 + p^2]^l [2p(1-p)]^{i-l} \right\}, \end{aligned}$$

where $2/n(n-1)$ is the probability that a read covers target SNP sites u and v; $(1-p)^2+p^2$ is the probability that a read covers SNPs that are identical given $h_u = h_v$; and the second summation (ranging from $\lfloor i/2 \rfloor + 1$ to i) represents for the majority voting operation evaluated over i voters.

Similarly, we have

$$\begin{aligned}\beta &\triangleq \Pr\{\text{majority voting claims } a_{uv}=1|h_u \neq h_v\}\\ &= \sum_{i=1}^{m}\left\{\binom{m}{i}\left[\frac{2}{n(n-1)}\right]^i\left[1-\frac{2}{n(n-1)}\right]^{m-i}\sum_{l=\lfloor i/2\rfloor+1}^{i}\binom{i}{l}[2p(1-p)]^l[(1-p)^2+p^2]^{i-l}\right\},\end{aligned}$$

where $2p(1-p)$ is the probability that a particular read covers SNPs that are identical given $h_u \neq h_v$. Since neither α nor β is straightforward to compute, we seek more compact and manageable bounds on these probabilities that will enable analysis of the worst-case scenarios.

Lemma 4. *When the number of reads used to assemble a long haplotype of length n scales as $m = \Theta(n\ln n)$, there exist positive constants κ_1, κ_2, and κ_3, such that*

$$\alpha \geq \frac{2\kappa_1\kappa_2[(1-p)^2+p^2]\ln n}{n-1}, \tag{18}$$

$$\beta \leq \frac{2\kappa_1[2p(1-p)]\ln n}{(n-1)(1-\kappa_3^{-1})}, \tag{19}$$

where $\kappa_2 < 1$ and $\kappa_3 > 1$.

The lemma shows that both α and β have bounds which scale as $\Theta(n^{-1}\ln n)$ (for the proof, please see Appendix B). Using these bounds, we next show that the signs of the corresponding entries of the eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{B}$ are identical with high probability.

C. Matrix Eigenvector Perturbation

After establishing the relationship between the adjacency matrix $\boldsymbol{A}$ and the planted model $\boldsymbol{B}$, we proceed to explore the difference between their eigenvectors by relying on the matrix perturbation theory. In particular, we show that for our choices of α and β, perturbation of the eigenvector of $\boldsymbol{A}$ associated with the second largest eigenvalue from the corresponding eigenvector of $\boldsymbol{B}$ (i.e., the difference between those two eigenvectors) vanishes as n increases. This result justifies performing spectral partitioning on $\boldsymbol{A}$, rather than $\boldsymbol{B}$, without a significant loss of performance.

The matrix perturbation theory allows one to determine sensitivity of matrix eigenvalues and eigenvectors with respect to slight perturbations. This area was pioneered in [11] where a general bound for the matrix eigenvalue perturbation effects was provided. More recently, [12] improved this bound under further assumptions on the matrix structure. Meanwhile, the famous Davis-Kahan sin-theta theorem [13] characterizes the rotation of eigenvectors after perturbation, and

[14] focuses on random matrices to propose a probabilistic sin-theta theorem. Note that the observed matrices in the haplotype assembly problem are always characterized by a particular structures, for instance, independent and binary distributed entries, low rank, etc. To exploit the special structure, we follow the result from a recent perturbation study [15] which provides a much tighter bound for the perturbation effects with respect to binary random matrices, summarized in the following lemma.

Lemma 5 (Lemma 2 and 3 in [15])**.** *Consider a square $n \times n$ symmetric 0-diagonal random matrix $\boldsymbol{M}$ such that its elements $m_{uv} = m_{vu}$ are independent Bernoulli random variables with parameters $\mathbb{E}[m_{uv}] = \rho_{uv}\chi n^{-1}$, where ρ_{uv} are constants and $\chi = \Omega(\ln n)$. Then, with probability at least $1 - O(n^{-1})$, we have*

$$|\lambda_k(\boldsymbol{M}) - \lambda_k(\mathbb{E}[\boldsymbol{M}])| \leq O(\chi^{1/2}), \tag{20}$$

$$||\boldsymbol{v}_k(\boldsymbol{M}) - \boldsymbol{v}_k(\mathbb{E}[\boldsymbol{M}])|| \leq O(\chi^{-1/2}), \tag{21}$$

for any k not larger than the rank of $\mathbb{E}[\boldsymbol{M}]$, where $\lambda_k(\boldsymbol{M})$ is the k-th largest eigenvalue of $\boldsymbol{M}$, and $\boldsymbol{v}_k(\boldsymbol{M})$ is the corresponding k-th eigenvector.

We observe that the adjacency matrix $\boldsymbol{A}$ has the same structure as the matrix $\boldsymbol{M}$ in the statement of the lemma. In particular, note that $\boldsymbol{A}$ is a 0-diagonal random matrix with each entry being an independently distributed Bernoulli random variable. The parameters of the Bernoulli distributions, α and β, satisfy the scale constraints with $\chi = \ln n$ due to Lemma 4. Moreover, note that $\mathbb{E}[\boldsymbol{A}] = \pi(\tilde{\boldsymbol{B}})$, where $\tilde{\boldsymbol{B}} = \boldsymbol{B} - \alpha\boldsymbol{I}$, and that permutation π does not change the eigenvectors. Therefore, we can utilize Lemma 5 to study the haplotype assembly problem. In particular, from (21) it follows that

$$||\boldsymbol{v}_2(\boldsymbol{A}) - \boldsymbol{v}_2(\tilde{\boldsymbol{B}})|| \leq O(\ln^{-1/2} n).$$

By noting that an addition of the identity matrix does not influence the eigenvectors, we conclude that $\boldsymbol{v}_2(\tilde{\boldsymbol{B}}) = \boldsymbol{v}_2(\boldsymbol{B})$. Thus, we obtain

$$||\boldsymbol{v}_2(\boldsymbol{A}) - \boldsymbol{v}_2(\boldsymbol{B})|| \leq O(\ln^{-1/2} n). \tag{22}$$

Recall that $\boldsymbol{v}_2(\boldsymbol{B})$ has the form of (15), which implies that a particular entry perturbed to change its sign contributes at least $\Omega(n^{-1/2})$ to $||\boldsymbol{v}_2(\boldsymbol{A}) - \boldsymbol{v}_2(\boldsymbol{B})||$. Therefore, if n_e is the number of errors, we have

$$\sqrt{\frac{n_e}{n}} \leq O(\ln^{-1/2} n). \tag{23}$$

By noting that n_e/n is the fraction of partition errors, we conclude that the haplotype can be recovered reliably with vanishing fraction of errors for sufficiently large number of reads n.

Remark 6. *As indicated by the analysis, spectral partitioning using SVD technique could only guarantee that the fraction of partition errors vanishes with high probability. For a stronger result, i.e., that the probability of partition error tends to zero, one may rely on another technique, "combinational projection" [10], instead of performing only the SVD. Essentially, the combinational projection gives another projection, after the one on the singular space, onto the span of characteristic vectors generated from a certain threshold. This way, the variances of target random variables are significantly reduced and the Chernoff-type argument could be adopted to arrive at a tighter bound on the distance of row spaces after the final projection. Note that* (20) *still holds in this case, and that by replacing the corresponding bounds in [10] it follows that* $\Theta(n \ln n)$ *reads are sufficient to exactly recover the haplotype with high probability.*

Remark 7. *Spectral partitioning is a very simple and computationally efficient algorithm that employs only the majority voting and the SVD techniques to perform haplotype assembly. In fact, we do not even require a full SVD calculation since only the second eigenvector is needed to determine the haplotype, as described in the algorithm. Therefore, by using the power method to discover the desired eigenvector, the complexity of spectral partitioning can be reduced from* $O(n^3)$ *in the general case to* $O(n \ln n)$ *for sparse adjacency matrix (since the number of total entries observed is roughly* $O(n \ln n)$*).*

Remark 8. *Although the theoretical analysis presented in Section IV is conducted under the assumption that there are precisely two entries observed in each row of the SNP fragment matrix, the results can easily be generalized to the case with multiple entries per row as long as reads may sample all pairs of SNP positions with non-trivial probability. If, however, the insert size is fixed or characterized by small variance, an alternative quantification of the minimum number of entries guaranteeing recovery of a low rank matrix may be needed. To this end, we note that a related matrix completion problem was studied in in [16] [17] [18], where an optimization approach was utilized to determine the necessary conditions and the recovery was facilitated* *by solving an appropriately formulated convex program. For our haplotype assembly problem, the observed fragments matrix $\boldsymbol{R}$ could be interpreted as a combination of the true haplotype matrix $\boldsymbol{S}$ and an independent sequencing error matrix $\boldsymbol{N}$. Moreover, the MEC criterion score is equivalent to the minimum l_1-norm of $\boldsymbol{N}$, and the associated optimization problem is given by*

$$\begin{aligned} min \quad & ||\boldsymbol{S}||_* + \gamma ||\boldsymbol{N}||_1 \\ s.t. \quad & \mathcal{P}_\Omega(\boldsymbol{S} \oplus \boldsymbol{N}) = \mathcal{P}_\Omega(\boldsymbol{R}), \end{aligned}$$

where $||\boldsymbol{S}||_*$ is the nuclear norm of $\boldsymbol{S}$ and γ denotes the balancing weight. [19] [20] report that the row space of the original matrix could be reliably recovered as long as the number of observed entries is large enough. Putting it more precisely, the number of reads needed for recovery is at least $\Omega(n \cdot poly(\ln n))$, which does not outperform the bound we obtained by relying on spectral partitioning. The kernel technique utilized in general for this type of proofs is the Golfing Scheme [19] [20], which requires a lower bound on the number of sampled entries to construct the dual certificate. If a new technique with a better performance guarantee could be used instead of the Golfing Scheme (at least for the case of the specific problem structure encountered in haplotype assembly), then the optimality method may also be able to match the necessary condition in scale. Results utilizing this optimization method will be reported elsewhere in the future.

V. Simulation Results and Analysis

A. Results on a Synthetic Data Set

We first test the performance of the two proposed algorithms – erasure decoding and spectral partitioning – on a synthetic data set. To this end, haplotypes are randomly generated according to a uniform distribution, followed by sampling paired-end fragments from haplotypes randomly and uniformly with i.i.d. sampling errors. For the moment, we enforce that 2 SNPs are observed in each fragment. The target of this simulation study is to empirically explore the relations among three key parameters featured in the algorithms, i.e., the length of the haplotype n, the probability of sampling errors p, and, most importantly, the number of sampled reads m. We show that the simulation results verify the conclusions of the theorems presented in the earlier sections of this paper, and also provide intuition for selecting appropriate parameters from the practical point of view.

To start, we set the probability of sampling error $p = 0.1$ (significantly larger than the typical value in practice), and study how the accuracy of haplotype assembly depends on relationship between the number of reads m and the haplotype length n. The results, shown in Fig. 4, provide the following observations:

- The erasure decoding algorithm fails to assemble the haplotype for all choices of m, which is basically due to large sampling noise. As indicated in Section III, this algorithm is intuitively designed for the error-free case and it has no performance guarantees when adopted and applied for the erroneous case.
- For spectral partitioning, choosing $m = \Theta(n)$ is not sufficient to ensure reliable recovery, while choosing $m = \Theta(n \ln n)$ is sufficient to guarantee that the error fraction vanishes for large n. This result is consistent with the conclusion of Theorem 2.
- Spectral partitioning, when implemented with sufficiently large number of reads (i.e., $m = \Theta(n \ln n)$), provides better error rate for large haplotype lengths. This is predicted by the theoretical result provided by equation (23).

Next, motivated by the results of the theoretical analysis and the previously described initial simulation results, we scale the number of sampled reads as $m = 2n \ln n$ and empirically study how the performance of both algorithms depends upon sampling errors and haplotype lengths. The results of simulation are illustrated in Fig. 5, leading to the following observations:

- The erasure decoding algorithm performs extremely well in the error-free case when the number of fragments is sufficiently large. However, in the erroneous case, this algorithm fails to recover the original haplotypes with high reliability.
- The convergence rate for spectral partitioning highly depends on p. More specifically, spectral partitioning is well-suited for the low-noise scenario, e.g., $p \leq 0.1$, which is typical of practical applications.

These results on synthetic data verify the results of our theoretical analysis in Section III and Section IV, and the overall conclusions may be summarized as follows:

1) Erasure decoding is applicable only in the noise-free setting and it requires $m = \Theta(n \ln n)$ reads for a reliable assembly of a haplotype of length n.
2) Spectral partitioning proves useful in the low-noise scenario (e.g., $p \leq 0.1$). It, too, requires $m = \Theta(n \ln n)$ reads for a reliable assembly of a haplotype of length n. When these two

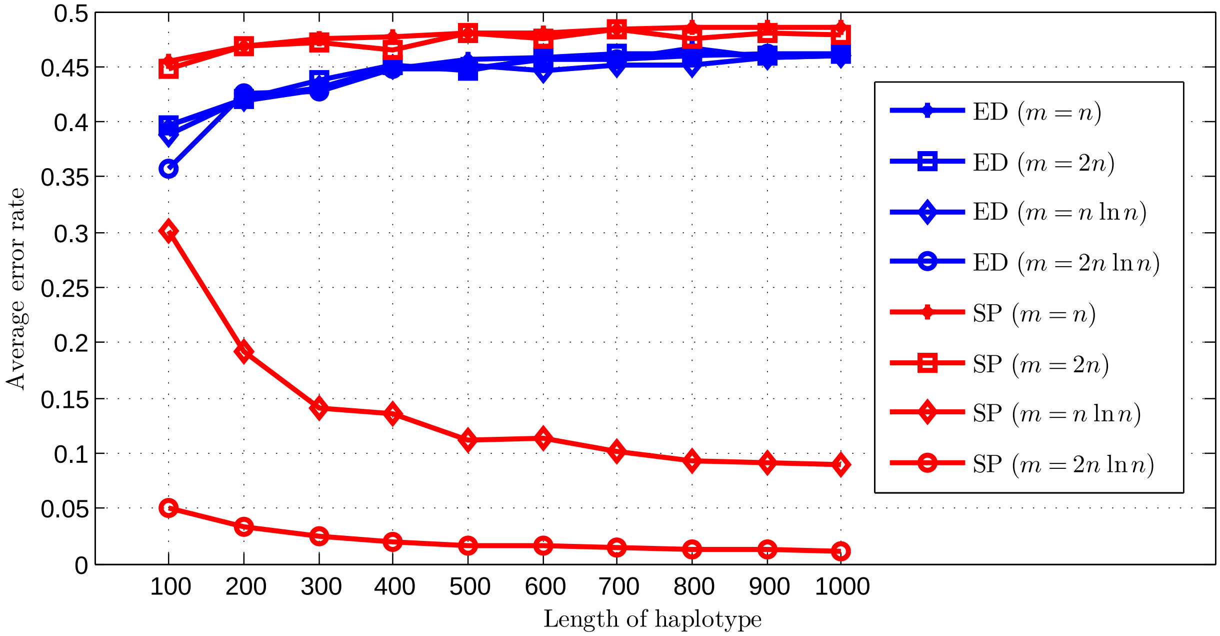

Fig. 4: Plot of average error rates from 100 random simulations where the probability of sampling errors is set to $p = 0.1$. In this simulation, we illustrate how the accuracy of haplotype assembly depends on relationship between the number of reads m and the haplotype length n for both erasure decoding (ED) and spectral partitioning (SP).

conditions are met, spectral partitioning is capable of recovering the original haplotype with high accuracy, and the recovery rate is inversely proportional to the length of the haplotype.

B. Test on a Benchmark Database

Here we present the study of the performance of both algorithms on the database created in [21], generated from the Phase I of the HapMap project [22] and widely adopted for benchmarking the effectiveness of haplotype assembly algorithms. This database consists of all 22 chromosomes from 209 unrelated individuals; shotgun sequencing process has been simulated to obtain the SNP observation matrix. Note that only heterogeneous SNP sites are considered in our study and that the recovery rate is computed based on the haplotype block lengths after filtering out the homozygous sites. Moreover, note that here the number of SNPs covered by reads varies and is no longer fixed to 2 as was the case in Section IV. Nevertheless, our algorithms

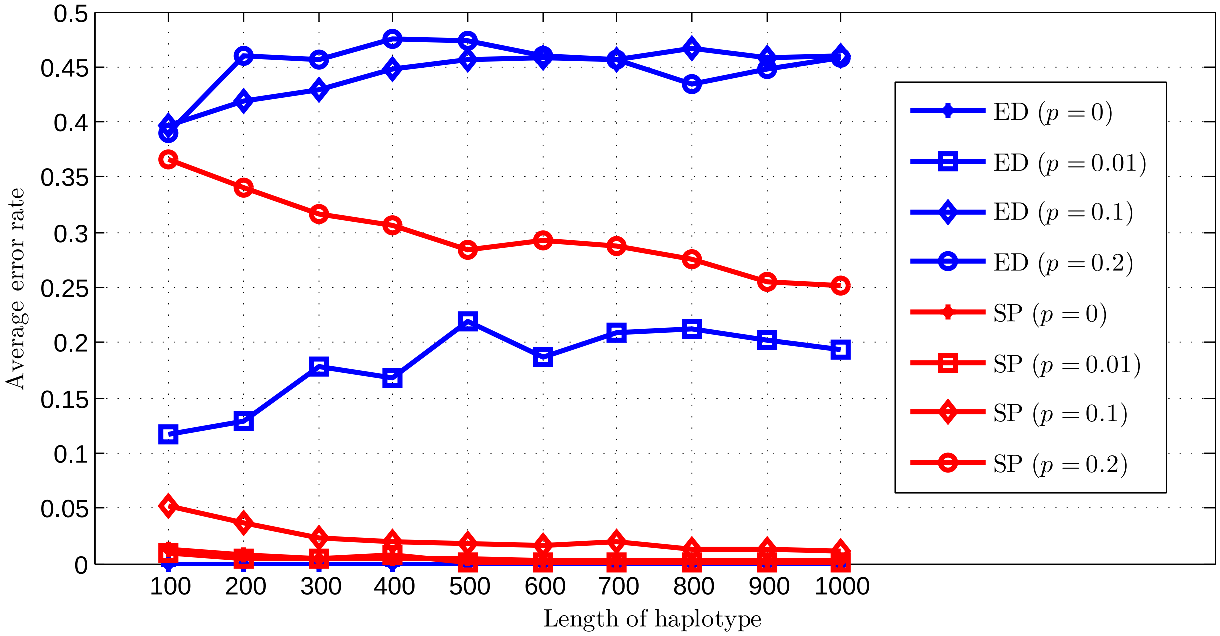

Fig. 5: Plot of the average error rates evaluated based on 100 random simulations where the number of reads is $m = 2n\ln n$. Here we illustrate how the performance depends on sampling errors for both erasure decoding (ED) and spectral partitioning (SP).

can be directly applied since the assumption on having precisely 2 observations per read was only needed to allow theoretical analysis.

TABLE I shows the average recovery rate computed using 100 data sets from [21], where the free parameters include: 1) the haplotype length $n = 100, 350, 700$; 2) the coverage $c = 3, 5, 8, 10$; and 3) the sampling error rate $p = 0\%, 10\%, 20\%$. From the simulation results, we find that erasure decoding successfully assembles the haplotype with high probability when $p = 0$, but fails to do so when $p > 0$. Moreover, sparse partitioning performs well in comparison with several recently proposed algorithms when the number of reads is sufficiently large. Therefore, our proposed algorithms, primarily meant to support theoretical results, also have practical significance.

VI. Conclusion

In this paper, we studied the haplotype assembly problem from an information-theoretic perspective. To determine the chromosome membership of reads provided by high-throughput sequencing systems and thus enable haplotype assembly, we interpret the problem as the one

Algorithms		$p = 0.0$				$p = 0.1$				$p = 0.2$			
		$c = 3$	$c = 5$	$c = 8$	$c = 10$	$c = 3$	$c = 5$	$c = 8$	$c = 10$	$c = 3$	$c = 5$	$c = 8$	$c = 10$
$n = 100$	SpeedHap	0.999	1.000	1.000	1.000	0.895	0.967	0.989	0.990	0.623	0.799	0.852	0.865
	Fast Hare	0.999	0.999	1.000	1.000	0.919	0.965	0.993	0.998	0.715	0.797	0.881	0.915
	2d-mec	0.990	0.997	1.000	1.000	0.912	0.951	0.983	0.988	0.738	0.793	0.873	0.894
	HapCUT	1.000	1.000	1.000	1.000	0.929	0.920	0.901	0.892	0.782	0.838	0.864	0.871
	MLF	0.973	0.992	0.997	0.998	0.889	0.970	0.985	0.995	0.725	0.836	0.918	0.938
	SHR-three	0.816	0.861	0.912	0.944	0.696	0.738	0.758	0.762	0.615	0.655	0.681	0.699
	DGS	1.000	1.000	1.000	1.000	0.930	0.985	0.989	0.997	0.725	0.813	0.878	0.917
	ED	1.000	1.000	1.000	1.000	0.650	0.651	0.627	0.639	0.587	0.581	0.585	0.593
	SP	0.958	0.997	0.999	1.000	0.883	0.961	0.990	0.995	0.687	0.809	0.918	0.943
$n = 350$	SpeedHap	0.999	1.000	1.000	1.000	0.819	0.959	0.984	0.984	0.439	0.729	0.825	0.855
	Fast Hare	0.990	0.999	1.000	0.999	0.871	0.945	0.985	0.995	0.684	0.746	0.853	0.877
	2d-mec	0.965	0.993	0.998	0.999	0.837	0.913	0.964	0.978	0.675	0.729	0.791	0.817
	HapCUT	1.000	1.000	1.000	1.000	0.930	0.913	0.896	0.888	0.771	0.831	0.862	0.867
	MLF	0.864	0.929	0.969	0.981	0.752	0.858	0.933	0.962	0.642	0.728	0.798	0.831
	SHR-three	0.830	0.829	0.895	0.878	0.682	0.724	0.742	0.728	0.591	0.632	0.670	0.668
	DGS	0.999	1.000	1.000	1.000	0.926	0.978	0.996	0.998	0.691	0.769	0.842	0.878
	ED	1.000	1.000	1.000	1.000	0.608	0.595	0.587	0.586	0.553	0.549	0.538	0.547
	SP	0.903	0.972	0.992	0.997	0.768	0.933	0.983	0.992	0.598	0.679	0.843	0.905
$n = 700$	SpeedHap	0.999	1.000	1.000	1.000	0.705	0.947	0.985	0.986	0.199	0.681	0.801	0.813
	Fast Hare	0.988	0.999	1.000	0.999	0.829	0.949	0.986	0.995	0.652	0.712	0.808	0.872
	2d-mec	0.946	0.976	0.992	0.997	0.786	0.880	0.948	0.965	0.647	0.697	0.751	0.778
	HapCUT	1.000	1.000	1.000	1.000	0.927	0.916	0.896	0.889	0.753	0.825	0.856	0.861
	MLF	0.787	0.854	0.919	0.933	0.698	0.809	0.863	0.884	0.624	0.682	0.747	0.765
	SHR-three	0.781	0.832	0.868	0.898	0.668	0.716	0.743	0.726	0.591	0.617	0.653	0.675
	DGS	0.999	1.000	1.000	1.000	0.931	0.977	0.987	0.997	0.669	0.741	0.818	0.861
	ED	1.000	1.000	1.000	1.000	0.576	0.571	0.572	0.573	0.534	0.532	0.531	0.528
	SP	0.887	0.967	0.991	0.997	0.723	0.910	0.977	0.990	0.562	0.610	0.751	0.843

TABLE I: Comparisons of our algorithms, erasure decoding (ED) and spectral partitioning (SP), with existing algorithms listed in [21]. Each entry in the table represents the average recovery rate from 100 randomly generated haplotype observation matrices, with respect to different n, c, and p.

of decoding data messages that are encoded and transmitted over a particular channel model. This channel model reflects the salient features of the paired-end sequencing technology and the haplotype assembly problem.

In the case of error-free sequencing, we find that the required number of reads needed for reconstruction is at least $\theta(n \ln n)$, where n denotes the length of the haplotype sequence. To establish a sufficient condition, we consider a graph interpretation of the original problem and analyze the erasure decoding algorithm that utilizes the common information across reads to iteratively recover haplotypes. We find that this algorithm ensures reconstruction with the optimal scaling of the number of reads.

In the case of erroneous sequencing, where errors are assumed to be generated independently and identically, we show that the number of reads needed to recover the haplotype is of the same order as in the error-free case. For the sufficient condition, we rephrase the original haplotype assembly problem as a low-rank matrix recovery. Using matrix permutation theory, we illustrate that haplotype sequences could be recovered reliably when the number of reads scales as $\Theta(n \ln n)$, where n denotes the haplotype length.

Simulation results corroborate theoretical claims, and the information-theoretic view of the haplotype assembly problem is worth pursuing in other applications (e.g., polyploid assembly).

Acknowledgements

The authors would like to thank Yudong Chen, Xinyang Yi, and Soyeon Ahn for insightful discussions. The authors also appreciate the reviewers' comments and suggestion, which help to improve the quality and depth of the paper.

References

[1] S. C. Schuster, *Nature*, vol. 200, no. 8, pp. 16–18, Jan. 2007.

[2] R. Schwartz, "Theory and algorithms for the haplotype assembly problem," *Communications in Information & Systems*, vol. 10, no. 1, pp. 23–38, Mar. 2010.

[3] P. J. Campbell, P. J. Stephens, E. D. Pleasance, S. O'Meara, H. Li, T. Santarius, L. A. Stebbings, C. Leroy, S. Edkins, C. Hardy *et al.*, "Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing," *Nature Genetics*, vol. 40, no. 6, pp. 722–729, Apr. 2008.

[4] A. Edwards, H. Voss, P. Rice, A. Civitello, J. Stegemann, C. Schwager, J. Zimmermann, H. Erfle, C. T. Caskey, and W. Ansorge, "Automated dna sequencing of the human hprt locus," *Genomics*, vol. 6, no. 4, pp. 593–608, Apr. 1990.

[5] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz, “SNPs problems, complexity, and algorithms,” in *9th Annual European Symposium (Algorithms - ESA 2001)*, Aarhus, Denmark, Aug. 2001, pp. 182–193.

[6] R. Lippert, R. Schwartz, G. Lancia, and S. Istrail, “Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem,” *Briefings in Bioinformatics*, vol. 3, no. 1, pp. 23–31, Mar. 2002.

[7] Z. Puljiz and H. Vikalo, “Decoding genetic variations: Communications-inspired haplotype assembly,” in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2015.

[8] A. S. Motahari, G. Bresler, and D. N. Tse, “Information theory of DNA shotgun sequencing,” *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6273–6289, Jun. 2013.

[9] V. Bansal and V. Bafna, “HapCUT: an efficient and accurate algorithm for the haplotype assembly problem,” *Bioinformatics*, vol. 24, no. 16, pp. i153–i159, Aug. 2008.

[10] F. McSherry, “Spectral partitioning of random graphs,” in *42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS 2001)*, Las Vegas, Nevada, USA, Oct. 2001, pp. 529–537.

[11] Z. Füredi and J. Komlós, “The eigenvalues of random symmetric matrices,” *Combinatorica*, vol. 1, no. 3, pp. 233–241, Sept. 1981.

[12] V. H. Vu, “Spectral norm of random matrices,” in *37th Annual ACM Symposium on Theory of Computing (STOC 2005)*, Baltimore, Maryland, USA, May 2005, pp. 423–430.

[13] C. Davis and W. M. Kahan, “The rotation of eigenvectors by a perturbation. III,” *SIAM Journal on Numerical Analysis*, vol. 7, no. 1, pp. 1–46, Mar. 1970.

[14] V. H. Vu, “Singular vectors under random perturbation,” *Random Structures & Algorithms*, vol. 39, no. 4, pp. 526–538, Dec. 2011.

[15] D. C. Tomozei and L. Massoulié, “Distributed user profiling via spectral methods,” *Stochastic Systems*, vol. 4, no. 0, pp. 1–43, 2014.

[16] E. J. Candès and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[17] B. Recht, “A simpler approach to matrix completion,” *The Journal of Machine Learning Research*, vol. 12, no. 104, pp. 3413–3430, Dec. 2011.

[18] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,” in *23th Annual Conference on Neural Information Processing Systems (NIPS 2009)*, Whistler, British Columbia, Canada, Dec. 2009, pp. 2080–2088.

[19] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *Journal of the ACM*, vol. 58, no. 3(11), pp. 1–37, May 2011.

[20] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, “Low-rank matrix recovery from errors and erasures,” *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4324–4337, Jul. 2013.

[21] F. Geraci, “A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem,” *Bioinformatics*, vol. 26, no. 18, pp. 2217–2225, Jul. 2010.

[22] International HapMap Consortium, “A haplotype map of the human genome,” *Nature*, vol. 437, no. 7063, pp. 1299–1320, Oct. 2005.

[23] P. Erdős and A. Rényi, “On random graphs I,” *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[24] B. Bollobás, *Random graphs*. Springer, 1998.

APPENDIX A
PROOF OF BOUNDS FOR ERROR-FREE CASE

Here, we first show a well-studied result for the lower and upper bounds on the connectivity of a random graph.

Lemma 9 ([23][24]). *Consider a random graph $G(n,q)$, where n is the number of nodes, and q is the probability of an edging included in the graph. Then, the graph is disconnected with probability at least $1-\varepsilon$, if $q<\frac{(1-\varepsilon)\ln n}{n}$; the graph is connected with probability at least $1-\varepsilon$, if $q>\frac{(1-\varepsilon)\ln n}{n}$, where ε is an arbitrarily small positive number.*

Therefore, $\ln n/n$ is a sharp threshold determining connectivity of a random graph. Considering the above lemma in the haplotype assembly setting, the probability that an edge is included to the graph can be calculated as

$$q=1-\left(1-\frac{2}{n(n-1)}\right)^m.$$

Hence, the threshold for connectivity becomes

$$\begin{aligned}m&=\frac{n(n-1)}{2}\cdot\frac{(1-\varepsilon)\ln n}{n}+O(n)\\&=\frac{1}{2}(1-\varepsilon)(n-1)\ln n+O(n),\end{aligned}$$

which gives $\Theta(n\ln n)$ for both lower and upper bounds with the scaling factor of $1/2$.

APPENDIX B
PROOF OF LEMMA 4

Assume $m=\kappa_1 n\ln n$, where κ_1 is a positive constant. In order to provide a lower bound for α, we truncate the first summation by leaving only the term with $i=1$. More precisely, we have

$$\begin{aligned}\alpha&=\sum_{i=1}^{m}\left\{\binom{m}{i}\left[\frac{2}{n(n-1)}\right]^i\left[1-\frac{2}{n(n-1)}\right]^{m-i}\sum_{l=\lfloor i/2\rfloor+1}^{i}\binom{i}{l}[(1-p)^2+p^2]^l[2p(1-p)]^{i-l}\right\}\\&\geq\binom{m}{1}\left[\frac{2}{n(n-1)}\right]\left[1-\frac{2}{n(n-1)}\right]^{m-1}\binom{1}{1}[(1-p)^2+p^2][2p(1-p)]^0\\&\geq\frac{2\kappa_1 n\ln n}{n(n-1)}e^{-\frac{4\kappa_1 n\ln n}{n(n-1)}}[(1-p)^2+p^2]\\&=\frac{2\kappa_1[(1-p)^2+p^2]n^{-\frac{4\kappa_1}{n-1}}\ln n}{n-1}.\end{aligned}$$

Note that $n^{-\frac{4\kappa_1}{n-1}}$ is an increasing function with n and tends to 1. Hence, for large enough n, there exists a constant $\kappa_2 < 1$ such that

$$n^{-\frac{4\kappa_1}{n-1}} \geq \kappa_2. \tag{24}$$

As a result, the lower bound becomes

$$\alpha \geq \frac{2\kappa_1 \kappa_2 [(1-p)^2 + p^2] \ln n}{n-1}. \tag{25}$$

Thus, α has a $\Theta(n^{-1} \ln n)$ scale lower bound. In fact, this bound is rather tight, because the first term ($i = 1$) dominates the overall value (analogue to the analysis of β that follows next).

In addition, we need to establish an upper bound on β. In particular, we show that the terms in the above summation are at least exponentially decreasing, such that the first term dominates the value of β. For this purpose, we denote

$$\beta_i \triangleq \binom{m}{i} \left[\frac{2}{n(n-1)}\right]^i \left[1 - \frac{2}{n(n-1)}\right]^{m-i} \sum_{l=\lfloor i/2 \rfloor+1}^{i} \binom{i}{l} [2p(1-p)]^l [(1-p)^2 + p^2]^{i-l}.$$

Introducing

$$\beta_i^{(l)} \triangleq \binom{i}{l} [2p(1-p)]^l [(1-p)^2 + p^2]^{i-l}$$

and

$$\beta = \sum_{i=1}^{m} \beta_i,$$

it follows that

$$\beta_i = \binom{m}{i} \left[\frac{2}{n(n-1)}\right]^i \left[1 - \frac{2}{n(n-1)}\right]^{m-i} \sum_{l=\lfloor i/2 \rfloor+1}^{i} \beta_i^{(l)}.$$

In order to derive a lower bound on β_i/β_{i+1} for any i, we focus on two cases:

1) For even i, write $i = 2k$ and note that

$$\begin{aligned}\frac{\beta_{2k}}{\beta_{2k+1}} &= \frac{\binom{m}{2k} \left[\frac{2}{n(n-1)}\right]^{2k} \left[1 - \frac{2}{n(n-1)}\right]^{m-2k} \sum_{l=k+1}^{2k} \beta_{2k}^{(l)}}{\binom{m}{2k+1} \left[\frac{2}{n(n-1)}\right]^{2k+1} \left[1 - \frac{2}{n(n-1)}\right]^{m-2k-1} \sum_{l=k+1}^{2k+1} \beta_{2k+1}^{(l)}} \\ &= \frac{(2k+1)[n(n-1)-2] \sum_{l=k+1}^{2k} \beta_{2k}^{(l)}}{2(\kappa_1 n \ln n - 2k) \sum_{l=k+1}^{2k+1} \beta_{2k+1}^{(l)}}.\end{aligned}$$

Note that there are $k+1$ terms for $\beta_{2k+1}^{(l)}$ in the denominator, but only k terms for $\beta_{2k}^{(l)}$ in the numerator. Hence, we duplicate the numerator to compare it with the denominator. More precisely, for $k+1 \leq l \leq 2k$,

$$\frac{\beta_{2k}^{(l)}}{\beta_{2k+1}^{(l)}} = \frac{2k+1-l}{(2k+1)[(1-p)^2+p^2]} \geq \frac{1}{2k+1}, \tag{26}$$

where the last inequality holds due to $(1-p)^2+p^2 \leq 1$. Moreover,

$$\frac{\beta_{2k}^{(k+1)}}{\beta_{2k+1}^{(2k+1)}} = \frac{(2k)![(1-p)^2+p^2]^{k-1}}{(k+1)!(k-1)![2p(1-p)]^k} \geq \frac{1}{2k+1}, \tag{27}$$

where the last inequality holds due to $1 \geq (1-p)^2+p^2 \geq 2p(1-p)$. Combining these two expressions, we have

$$\begin{aligned}
\frac{2\beta_{2k}}{\beta_{2k+1}} &= \frac{(2k+1)[n(n-1)-2]\left\{\sum\limits_{l=k+1}^{2k} \beta_{2k}^{(l)} + \sum\limits_{l=k+1}^{2k} \beta_{2k}^{(l)}\right\}}{2(\kappa_1 n \ln n - 2k)\left\{\sum\limits_{l=k+1}^{2k} \beta_{2k+1}^{(l)} + \beta_{2k+1}^{(2k+1)}\right\}} \\
&\geq \frac{(2k+1)[n(n-1)-2]\left\{\sum\limits_{l=k+1}^{2k} \beta_{2k}^{(l)} + \beta_{2k}^{(k+1)}\right\}}{2(\kappa_1 n \ln n - 2k)\left\{\sum\limits_{l=k+1}^{2k} \beta_{2k+1}^{(l)} + \beta_{2k+1}^{(2k+1)}\right\}} \\
&\geq \frac{(2k+1)[n(n-1)-2]}{2(\kappa_1 n \ln n - 2k)(2k+1)} \\
&= \frac{n(n-1)-2}{2(\kappa_1 n \ln n - 2)}.
\end{aligned}$$

Thus,

$$\frac{\beta_{2k}}{\beta_{2k+1}} \geq \frac{n(n-1)-2}{4(\kappa_1 n \ln n - 2)}. \tag{28}$$

2) For i odd, write $i = 2k-1$ and note that

$$\begin{aligned}
\frac{\beta_{2k-1}}{\beta_{2k}} &= \frac{\binom{m}{2k-1}\left[\frac{2}{n(n-1)}\right]^{2k-1}\left[1-\frac{2}{n(n-1)}\right]^{m-2k+1}\sum\limits_{l=k}^{2k-1}\beta_{2k-1}^{(l)}}{\binom{m}{2k}\left[\frac{2}{n(n-1)}\right]^{2k}\left[1-\frac{2}{n(n-1)}\right]^{m-2k}\sum\limits_{l=k+1}^{2k}\beta_{2k}^{(l)}} \\
&= \frac{2k[n(n-1)-2]\sum\limits_{l=k}^{2k-1}\beta_{2k-1}^{(l)}}{2(\kappa_1 n \ln n - 2k+1)\sum\limits_{l=k+1}^{2k}\beta_{2k}^{(l)}}.
\end{aligned}$$

In this case, both numerator and denominator have k terms in summation. Hence, term-by-term comparison leads to

$$\frac{\beta_{2k-1}^{(l)}}{\beta_{2k}^{(l)}} = \frac{2k-l}{2k[(1-p)^2+p^2]} \geq \frac{1}{2k}. \tag{29}$$

Thus,

$$\frac{\beta_{2k-1}}{\beta_{2k}} \geq \frac{n(n-1)-2}{2(\kappa_1 n \ln n - 1)}. \tag{30}$$

Note that, in both cases, the lower bounds (28) and (30) tend to infinity as n increases. Therefore, there exists a constant $\kappa_3 > 1$ such that for large enough n

$$\min\left\{\frac{n(n-1)-2}{4(\kappa_1 n \ln n - 2)}, \frac{n(n-1)-2}{2(\kappa_1 n \ln n - 1)}\right\} \geq \kappa_3, \tag{31}$$

which further implies that for any value of i,

$$\frac{\beta_i}{\beta_{i+1}} \geq \kappa_3.$$

Based on this we obtain $\beta_i \leq \beta_1 \kappa_3^{1-i}$ and

$$\begin{aligned}
\beta_1 &= \binom{m}{1}\left[\frac{2}{n(n-1)}\right]\left[1-\frac{2}{n(n-1)}\right]^{m-1}\binom{1}{1}[2p(1-p)][(1-p)^2+p^2]^0 \\
&\leq \frac{2\kappa_1 n \ln n}{n(n-1)} e^{-\frac{2\kappa_1 n \ln n}{n(n-1)}}[2p(1-p)] \\
&= \frac{2\kappa_1[2p(1-p)]n^{-\frac{2\kappa_1}{n-1}}\ln n}{n-1} \\
&\leq \frac{2\kappa_1[2p(1-p)]\ln n}{n-1},
\end{aligned}$$

where we have used the fact that $n^{-\frac{2\kappa_1}{n-1}} \leq 1$. Hence, we obtain

$$\begin{aligned}
\beta &= \sum_{i=1}^{m} \beta_i \\
&\leq \sum_{i=1}^{m} \beta_1 \kappa_3^{1-i} \\
&\leq \frac{2\kappa_1[2p(1-p)]\ln n}{(n-1)(1-\kappa_3^{-1})}.
\end{aligned} \tag{32}$$

Thus, the upper bound for β is also $\Theta(n^{-1}\ln n)$ scale.

A point to clarify: in several places we have somewhat imprecisely used categorization "large enough n". One may be concerned with whether a particular choice of n satisfying the proof assumptions could match the practical haplotype assembly scenarios. As an illustration, for $\kappa_1 = 2$ that we use in the simulation setup, a simple choice of $\kappa_2 = 1/2$ and $\kappa_3 = 2$ implies that the minimum value of n needed to satisfy both assumptions (24) and (31) is given by

$$n \geq \max\{45, 69, 28\} = 69,$$

which is quite smaller than the commonly encountered value in the haplotype assembly problems. Therefore, our bounds are meaningful and useful in practical scenarios.