Optimal Haplotype Assembly with Statistical Pruning

Shreepriya Das
ECE Department
The University of Texas at Austin
Austin, USA
shreedas@utexas.edu

Haris Vikalo
ECE Department
The University of Texas at Austin
Austin, USA
hvikalo@ece.utexas.edu

Abstract—Solving the haplotype assembly problem by optimizing the commonly used minimum error correction criterion is known to be NP-hard. For this reason, suboptimal heuristics are often used in practice. In this paper, we propose a novel method for optimal haplotype assembly that is based on depth-first branch-and-bound search of the solution space. Our scheme is inspired by the sphere decoding algorithms used heavily in the field of digital communications. Using the statistical information about errors in sequencing data, we constrain the search of the haplotype space and speedily find the optimal solution to the haplotype assembly problem. Theoretical analysis of the expected complexity of the algorithm shows that optimal haplotype assembly is practically feasible for haplotype blocks of moderate lengths typically obtained using present day high throughput sequencers. The scheme is then tested on 1000 Genomes Project experimental data to verify the efficacy of the proposed method.

I. INTRODUCTION

The complete information about DNA variations in an individual genome is given by haplotypes, the ordered list of alleles on a single chromosome. Haplotype information is of fundamental importance for a wide range of applications. These include the discovery of an individual's susceptibility to diseases and response to therapeutic drugs [1], whole genome association studies on *tag SNPs* (as in HapMap project [2]), and understanding of recombination patterns and identification of genes under positive selection [3].

Haplotypes of an individual whose genome is sequenced can be assembled using short reads provided by high-throughput sequencing platforms. Each read represents a fragment of one chromosome of the individual. The SNP rate between two human haploid chromosomes (*i.e.*, two chromosomes in a homozygous pair) is estimated based on a panel of individuals to be $\sim 10^{-3}$ [4]. Since next-generation sequencing reads are relatively short (the latest platforms are capable of producing reads of a few hundred bases in length), they are unlikely to cover multiple SNP sites. This challenge has lately been ameliorated with the emergence of the paired-end technology where we sequence pairs of fragments located at opposite ends of a DNA strand of known length. The pairs of reads may be separated by several thousands of bases, enabling us to

link haplotype information over long distances and thus enable their reconstruction.

In practice, however, sequenced reads are erroneous, thus making haplotype assembly difficult. Various formulations of the haplotype assembly problem have been proposed [5]. In this paper, we focus on the minimum error correction (MEC) formulation, which poses the assembly problem as one of finding the smallest number of nucleotides in reads whose flipping to a different value would resolve error-induced conflicts among the fragments originating from the same chromosome. Finding the optimal solution to the MEC formulation of the haplotype assembly problem is known to be NP-hard [5].

A number of heuristic haplotype assembly methods have been proposed recently. In [6], a greedy algorithm was applied to the first complete diploid individual genome obtained via high-throughput sequencing. [7] (HapCUT) used a max-cut formulation of the problem to significantly improve on the performance of the greedy scheme. [8] (HASH) and [9] relied on MCMC and Gibbs sampling schemes to tackle the same problem. RefHap [10], relying on a greedy cut approach, was recently introduced and applied to reads sequenced using fosmid libraries, while HapCompass [11] relied on a graphical approach to develop a scheme which resolves conflicts arising from incorrect haplotype phasing. In addition to the above heuristics, several computationally intensive methods for finding exact solutions to various formulations of the haplotype assembly problem have been proposed in literature. In [12], the authors used a branch-and-bound scheme to minimize the MEC objective over the space of reads, imposing a bound on the objective obtained by a random bipartition of the reads. Unfortunately, exponential growth of the complexity of this scheme renders it computationally too expensive even for moderate haplotype lengths.

Motivated by the so-called sphere decoding algorithm from the field of data communications [13], [14], we propose a depth-first branch-and-bound scheme which uses statistical information about sequencing errors to impose an upper bound on the objective function and facilitate computationally efficient search for the optimal solution to the MEC formulation of the haplotype assembly problem. We analyze the expected complexity of the proposed algorithm and show that it is practically feasible for moderate lengths of haplotype blocks.

This work was supported in part by the National Science Foundation under grant CCF-1320273.

II. HAPLOTYPE ASSEMBLY WITH STATISTICAL PRUNING

After mapping the reads obtained from a sequencing platform to the reference genome, nucleotides in heterozygous sites of homologous chromosomes are labeled as -1 or 1 according to a selected convention. Since homozygous sites do not cause any ambiguity in haplotype assembly, they are omitted from the haplotype and read representations. Thus a haplotype pair can be represented as an unordered pair of strings $(\mathbf{h}^1, \mathbf{h}^2)$, each of length n , with components $h_i^1, h_i^2 \in \{1, -1\}$. Since $\mathbf{h}^1$ and $\mathbf{h}^2$ are complements of each another, we simplify the notation and introduce $\mathbf{h} = \mathbf{h}^1 = -\mathbf{h}^2$. Consequently, each read can now be represented as a ternary string of length n with entries $\{-1, 0, 1\}$, where 0's indicate SNP positions on the chromosome that are not covered by the read. The reads are arranged into an $m \times n$ matrix R according to their positions along the chromosome, where m denotes the number of reads. In particular, the i^{th} row of R , r_i , corresponds to the i^{th} read, and has entries -1 or 1 in locations corresponding to heterozygous positions on the chromosome covered by the read. The start and the end of the i^{th} read are the first and last position in r_i that are not 0. Note that paired-end reads typically contain substrings comprising all 0's which correspond to gaps connecting paired-end fragments; such substrings may also be present in single reads if there is missing data. Reads with a continuous string of $+1$ and -1 are gapless reads, otherwise they are referred to as gapped reads. The length of a read starting at position i and ending at position j is $j - i + 1$ (i.e., it includes any possible gaps).

We should point out that certain pre-processing steps need to be performed prior to formulating and solving the haplotype assembly problem. Note that a read overlaps two SNP positions if the entries of the read at those positions in the matrix are non-zero. Reads that cover only one SNP position are eliminated. Similarly, any SNP position not covered by at least one read is removed. Furthermore, from R we can generate an adjacency matrix and a graph having vertices that correspond to the SNP positions (i.e., to the columns of R). An edge is present between two vertices if a read overlaps the corresponding SNP positions. After forming the adjacency matrix, disconnected subgraphs or partitions are extracted from it, effectively partitioning the matrix into smaller disconnected matrices.

Let us define a measure of the distance d between two symbols in the ternary alphabet as

$$d(x, y) = \begin{cases} 1 & \text{if } x \neq 0 \text{ and } y \neq 0 \text{ and } x \neq y, \\ 0, & \text{otherwise.} \end{cases}$$

Denote the Hamming distance between read r_i and haplotype $\mathbf{h}$ as $\text{hd}(r_i, \mathbf{h}) = \sum_{j=1}^n d(r_{i,j}, h_j)$. Then the minimum error criterion (MEC) formulation of the haplotype assembly problem is concerned with minimizing Z over $\mathbf{h}$, where the objective function

$$Z = \sum_{i=1}^m \min(\text{hd}(r_i, \mathbf{h}), \text{hd}(r_i, -\mathbf{h})), \quad (1)$$

and m denotes the total number of reads.

To minimize Z in (1) over $\mathbf{h}$, we construct and search a binary tree where the node at the k^{th} level of the tree corresponds to the partial (k bases long) leading substring of $\mathbf{h}$. Recall that conflicts between reads are induced by sequencing errors and that in the MEC framework we inherently attempt to resolve conflicts by assuming the fewest possible sequencing errors. For simplicity, we here assume a flat error profile of the reads and denote the probability of mis-calling a base by p (where p is approximately one-third of the sequencing error rate). The total number of errors v causing conflicts between fragments has a binomial distribution with cumulative mass function (cmf)

$$\mathbb{P}(v \leq k) = \sum_{i=0}^k \binom{\mu}{i} (p)^i (1-p)^{(\mu-i)},$$

where μ denotes the total number of non-zero entries in matrix R . This, in turn, can be used to impose a statistical bound on the objective function (1). (A similar idea has been used to enable efficient search for the closest point in a lattice in a probabilistic setting commonly encountered in data communications [13], [14].) In particular, we use a depth-first search on the aforementioned binary tree to find all haplotype candidates $\mathbf{h}$ satisfying

$$Z = \sum_{i=1}^m \min(\text{hd}(\mathbf{r}_i, \mathbf{h}), \text{hd}(\mathbf{r}_i, -\mathbf{h})) \leq b, \quad (2)$$

where the upper bound on the number of errors, b , is computed from the binomial cmf as

$$b = \inf\{k \mid \sum_{i=0}^k \binom{\mu}{i} (p)^i (1-p)^{(\mu-i)} > (1-\varepsilon)\}. \quad (3)$$

Parameter ε determines the confidence that we will find an n -dimensional $\mathbf{h}$ which satisfies the constraint (2). For instance, setting $\varepsilon = 0.01$ means that with probability $1 - \varepsilon = 0.99$ we will find a solution to the constraint satisfaction problem (2). Clearly, while traversing close to the root of the search tree, the bound is very loose and induces little pruning. As we proceed deeper into the search tree, the bound is more frequently violated which results in discarding a large fraction of the nodes. If no $\mathbf{h}$ satisfying (2) is found, we increase the bound (i.e., reduce ε) and start the search anew. Every time a candidate $\mathbf{h}^c$ satisfying (2) is found, we can update the bound by setting $b = \sum_{i=1}^m \text{hd}(\mathbf{r}_i, \mathbf{h}^c)$ and proceed searching.

III. EXPECTED COMPLEXITY OF THE ALGORITHM

The expected complexity of the proposed haplotype assembly algorithm is proportional to the average number of nodes visited in the search tree. In particular, the number of tree nodes that survive the pruning process is given by

$$N_p = \sum_{j=2}^n \sum_{k=1}^{2^j} \mathbb{1}_{\{C_k^j \leq b\}}, \quad (4)$$

where C_k^j denotes the cost associated with inclusion of the k^{th} node at level j and b is the statistically determined upper bound on the value of the objective function. More precisely, the “cost” is the total MEC score computed for the partially reconstructed haplotype sequence. The expected number of survivor nodes is then given by the expectation of (4), i.e.,

$$\mathbb{E}\{N_p\} = \sum_{j=2}^n \sum_{k=1}^{2^j} \mathbb{P}(C_k^j \leq b), \quad (5)$$

and the expected complexity is given by

$$\mathbb{E}\{C_p\} = \sum_{j=2}^n \sum_{k=1}^{2^j} \mathbb{P}(C_k^j \leq b) f_p,$$

where f_p is the number of elementary operations that need to be performed per each visited node.

A. Expected number of visited tree points

To find the expected number of points visited in the search tree, we here consider the setting where the length of a haplotype block is n , there are m reads and the position of the l non-zero elements in the read r_i are arbitrary. Moreover, we also assume that the 2 haplotype strands are sampled with equal probability, i.e., any read is equally likely originating from either of the chromosomes in a pair. For the sake of complexity analysis, assume (without a loss of generality) that the true haplotype is $(1, \dots, 1), (-1, \dots, -1)$. At level j , there are 2^{j-1} nodes that are candidates for a partially assembled haplotype. The contributions of individual reads to the value of the objective function are mutually independent, and hence the probability mass function of C_k^j can be obtained by convolving probability mass functions modeling random contributions of individual reads to C_k^j . Rather than evaluating convolutions, which turns out to be somewhat cumbersome, we turn our attention to the moment generating functions (MGF) of the distributions; recall that the MGF of a sum of random variables is equal to the product of the MGF of the individual variables.

Omitting details for brevity, it suffices to say that the MGF of C_k^j can be found as

$$\frac{1}{2^j} \sum_{k=0}^{2^j} \prod_{r=1}^m p_0^r + p_1^r e^t + p_2^r e^{2t} \dots = p_{\lfloor \frac{1}{2} \rfloor}^r e^{\lfloor \frac{1}{2} \rfloor t},$$

where p_i^r are coefficients in the evaluated MGF. As an illustration of the MGF computation, consider the scenario where the reads are of length 2 and focus on the node $(1, -1), (-1, 1)$. Each read contributes to the MEC score either 0 or 1. The node incurs zero cost if there is exactly 1 error in the read. Since there are 2 positions at which this may occur, $\mathbb{P}(C = 0) = 2p(1-p) = p'$ and $\mathbb{P}(C = 1) = (1-p)^2 + p^2 = 1 - 2p(1-p)$. Therefore, this is simply a Bernoulli random variable. For the other node, $(1, 1), (-1, -1)$, zero cost is incurred if there are no errors or if there are 2 errors. Thus $\mathbb{P}(C = 0) = (1-p)^2 + p^2$, $\mathbb{P}(C = 1) = 2p(1-p)$, and (p_0, p_1) is node dependent. The distribution of C_k^2 (partial MEC score

Fig. 1. Plots comparing theoretical with empirical expected complexity exponent. Top: The complexity exponent as a function of tree levels. Bottom: The complexity exponent as a function of haplotype block length.

evaluated at 2nd level of the search tree) is thus a mixture of 2 binomial distributions,

$$C_k^2 \sim \frac{1}{2} (B(c, p') + B(c, 1 - p')),$$

where $B(c, p')$ is the binomial distribution with parameters c and p' , and c is the coverage. The corresponding MGF is

$$\frac{1}{2} [((1 - p') + p' e^t)^c + (p' + (1 - p') e^t)^c]$$

We evaluate the MGF numerically and read out their coefficients to determine $\mathbb{P}(C_k^j \leq b)$ and, subsequently, evaluate (5). Such numerical evaluation of the expected complexity is computationally inefficient since for a haplotype of block length n , one needs to evaluate the contributions from the MGFs of 2^n nodes. Under some mild assumptions, we can further improve the efficiency of calculating the MGF and, hence, the expected complexity of the algorithm (details omitted for brevity).

In Fig. 1, we compare the theoretical and empirical complexity exponent for different design instances. The complexity exponent is defined as $\log(C_p)/\log(n)$, where C_p is the total (expected) complexity and n is the length of haplotype. The parameters of the study are $p = \frac{0.01}{3}$ (one third of the base-calling error rate), $c = 10$ (coverage), haplotype block length

$= 75$, $l = 2$. For our various plots we keep all parameters fixed except the parameter being studied. The top plot in Fig. 1 shows the complexity exponent as a function of the level. The bottom plot in Fig. 1 demonstrates the growth of the complexity exponent as a function of length. As can be seen there, theoretical results closely match empirical ones. Moreover, the complexity exponent is sub-quadratic over the considered range of haplotype lengths, implying practical feasibility of the optimal haplotype assembly using the proposed algorithm.

IV. RESULTS ON 1000 GENOMES PROJECT DATA

In addition to the simulation studies presented in the previous section, we tested our algorithm on all 22 chromosomes of an individual (NA12878) in the 1000 Genomes Project data set. Table I shows the performance in terms of the MEC of our proposed algorithm and that of HapCUT (the latter is the widely used heuristic shown to perform well in various setups). As can be seen in this table, the MEC score obtained with our proposed algorithm is better than that of HapCUT.

chr #	MEC	
	Statistical Pruning	HapCUT
1	2306	2317
2	2872	2887
3	2361	2367
4	2605	2618
5	2167	2179
6	3559	3576
7	2070	2077
8	1838	1855
9	1479	1489
10	1823	1828
11	1577	1583
12	1589	1591
13	1405	1414
14	987	992
15	1061	1062
16	1263	1271
17	1228	1236
18	941	942
19	765	767
20	794	797
21	528	533
22	436	438

TABLE I

The MEC of the proposed method and HapCUT haplotype assembly strategies for 1000 Genomes Project data. The novel algorithm outperforms HapCUT on all chromosomes.

V. CONCLUSIONS

In this paper, we developed a novel depth-first branch-and-bound haplotype assembly algorithm that relies on statistical information about sequencing errors to efficiently find the optimal solution to the assembly problem. We demonstrated theoretically that the expected complexity of the algorithm is low and that the optimal assembly of haplotype blocks of moderate length is practically feasible. Benchmarking on 1000 Genomes Project demonstrates the superior performance of the proposed method as compared to the widely used

heuristic algorithm HapCut. As a part of future work, pursuing modifications of the statistical bound which induce more aggressive pruning while searching near the root of the tree but potentially sacrifice optimal performance is of interest.

REFERENCES

- [1] A. G. Clark, "The role of haplotypes in candidate gene studies," *Genetic epidemiology*, vol. 27, no. 4, pp. 321–333, 2004.
- [2] R. A. Gibbs, J. W. Belmont, P. Hardenbol, T. D. Willis, F. Yu, H. Yang, L.-Y. Ch'ang, W. Huang, B. Liu, Y. Shen *et al.*, "The international hapmap project," *Nature*, vol. 426, no. 6968, pp. 789–796, 2003.
- [3] P. C. Sabeti, D. E. Reich, J. M. Higgins, H. Z. Levine, D. J. Richter, S. F. Schaffner, S. B. Gabriel, J. V. Planko, N. J. Patterson, G. J. McDonald *et al.*, "Detecting recent positive selection in the human genome from haplotype structure," *Nature*, vol. 419, no. 6909, pp. 832–837, 2002.
- [4] R. Sachidanandam, D. Weissman, S. C. Schmidt, J. M. Kakol, L. D. Stein, G. Marth, S. Sherry, J. C. Mullikin, B. J. Mortimore, D. L. Willey *et al.*, "A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms," *Nature*, vol. 409, no. 6822, pp. 928–933, 2001.
- [5] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz, "Snps problems, complexity, and algorithms," in *AlgorithmsESA 2001*. Springer, 2001, pp. 182–193.
- [6] S. Levy, G. Sutton, P. C. Ng, L. Feuk, A. L. Halpern, B. P. Walenz, N. Axelrod, J. Huang, E. F. Kirkness, G. Denisov *et al.*, "The diploid genome sequence of an individual human," *PLoS biology*, vol. 5, no. 10, p. e254, 2007.
- [7] V. Bansal and V. Bafna, "Hapcut: an efficient and accurate algorithm for the haplotype assembly problem," *Bioinformatics*, vol. 24, no. 16, pp. i153–i159, 2008.
- [8] V. Bansal, A. L. Halpern, N. Axelrod, and V. Bafna, "An MCMC algorithm for haplotype assembly from whole-genome sequence data," *Genome research*, vol. 18, no. 8, pp. 1336–1346, 2008.
- [9] J. H. Kim, M. S. Waterman, and L. M. Li, "Diploid genome reconstruction of ciona intestinalis and comparative analysis with ciona savignyi," *Genome research*, vol. 17, no. 7, pp. 1101–1110, 2007.
- [10] J. Duitama, G. K. McEwen, T. Huebsch, S. Palczewski, S. Schulz, K. Verstrepen, E.-K. Suk, and M. R. Hoehe, "Fosmid-based whole genome haplotyping of a hapmap trio child: evaluation of single individual haplotyping techniques," *Nucleic acids research*, p. gkr1042, 2011.
- [11] D. Aguiar and S. Istrail, "Hapcompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data," *Journal of Computational Biology*, vol. 19, no. 6, pp. 577–590, 2012.
- [12] R.-S. Wang, L.-Y. Wu, Z.-P. Li, and X.-S. Zhang, "Haplotype reconstruction from snp fragments by minimum error correction," *Bioinformatics*, vol. 21, no. 10, pp. 2456–2462, 2005.
- [13] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm i. expected complexity," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2806–2818, 2005.
- [14] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm ii. generalizations, second-order statistics, and applications to communications," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2819–2834, 2005.