COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 10, No. 1, pp. 23-38, 2010

002

THEORY AND ALGORITHMS FOR THE HAPLOTYPE ASSEMBLY PROBLEM*

RUSSELL SCHWARTZ†

Abstract. Genome sequencing studies to date have generally sought to assemble consensus genomes by merging sequence contributions from multiple homologous copies of each chromosome. With growing interest in genetic variations, however, there is a need for methods to separate these distinct contributions and assess how individual homologous chromosome copies differ from one another. An approach to this problem was developed using small sequence fragments derived from shotgun sequencing studies to determine the patterns of variations that co-occur on individual chromosomes. This has become known as the "haplotype assembly" problem. This review paper surveys results on the theory and algorithms for haplotype assembly. It first describes common abstractions of the problem. It then discusses some notable intractibility results for different problem variants. It next examines a variety of combinatorial, statistical, and heuristic methods for assembling fragment data sets in practice. The review concludes with a discussion of recent directions in diploid genome sequencing and their implications for haplotype assembly in the future.

1. Introduction. The availability of the first consensus human genome sequences [26, 28] has spawned numerous lines of research into genome function, organization, and evolution. One major avenue of work to arise from genome sequencing is the study of genome variations, or polymorphisms, which describe how individual copies of the genome differ among members of a single species. Several major projects are underway to characterize and catalog the common forms of genetic variation in the human genome and how they are distributed among individuals, populations, and even individual chromosomes within one person [23, 27, 21] . Particularly notable among these has been the International Haplotype Map (HapMap) project [24, 25], which is in the process of gathering genotype sequences from diverse human populations in order to assess the nature and distribution of variation between homologous chromosomes across the human species. A key step in this analysis is determining likely *haplotypes*, each describing the sequence of a portion of a single chromosome copy in a single individual. While haplotype determination can be accomplished painstakingly through direct sequencing of cloned data, it is more commonly and efficiently done by computational inference. Data are generally sequenced in a diploid form known as a *genotype*, where contributions from both chromosomes in a homologous pair are conflated, and then computer algorithms are used to infer likely haplotypes from these diploid genotypes [6]. For a review of this computational inference problem, known as *haplotype phasing*, the reader can refer to Halldórsson et al. [10].

Haplotype assembly (sometimes called "individual haplotyping" or "single indi-

*Dedicated to Michael Waterman on the occasion of his 67th birthday.

†Department of Biological Sciences and Lane Center for Computational Biology, Carnegie Mellon University, 4400 Fifth Avenue, Pittsburgh PA 15213, E-mail: russells@andrew.cmu.edu

vidual haplotyping") emerged as an alternative to the inference of haplotypes from genotypes. In a haplotype assembly approach, one constructs haplotypes from short haploid *fragments* of DNA, each representing a small sequenced piece of one chromosome. Haplotype assembly was first developed as a fortuitous side-effect of the Celera human genome sequencing project [28], whose shotgun sequencing approach [8] involved generating several million haploid DNA fragments, each drawn from a pool of five donors. While most sequenced bases on a fragment will be identical regardless of its chromosome of origin, each fragment's sequence may include the sequence of one or more variant sites. We say that a fragment *covers* those sites for which it has sequence data. These haploid fragments provided a data source from which one can in principle directly identify haplotypes, provided one can find a way to sort them into their source chromosomes and assemble them into sufficiently long stretches of sequence. This sorting is itself a difficult inference problem, however, because most fragments are approximately 1 kilobase (kb), roughly comparable to the distance between variant sites on two homologous chromosomes. A fragment is therefore unlikely to cover more than one DNA site that varies between the two chromosome copies. It is thus unlikely to directly carry any information from which one can determine which variants co-occur on the same chromosome (a process known as *phasing* the data).

The phasing problem becomes feasible in part because Celera data was predominantly sequenced in the form of *mate pairs*, pairs of short sequences that are derived from opposite ends of a single DNA strand of known length [7, 33]. While the two paired ends were limited by the sequencing technology to about 1 kb, the distance between each pair can be many kilobases. These mate pairs make it possible to link phase information over long regions of sequence and thus potentially to construct non-trivial haplotypes. Mate-paired sequencing, also known as *paired-end sequencing* or *double-barrel shotgun sequencing*, is illustrated in Fig. 1, which shows how a set of mate-paired fragments might be derived from a pair of homologous chromosomes. The result is a set of fragment sequences, each covering some possible discontinuous set of variable sites, which are the input to a haplotype assembly problem. For purposes of illustration, we assume these sites take the form of single nucleotide polymorphisms (SNPs), at which a single DNA base differs between the two DNA strands. In practice, though, the methods described here can be applied to any form of variation.

The problem has since attracted attention from several research communities. Initially, the primary goal of this work was modest: to group fragments into pairs of chromosomes to identify those fragments that were mistakenly assigned to an incorrect but similar region of the genome (known as *paralogous misrecruitment*). It was also useful as a way of discovering specific haplotypes that might be used to seed genotype-based phase inference algorithms in order to improve their accuracy. In these contexts, the problem attracted a research community among computational biologists who have formulated several problem variants [14, 19], produced a variety of theoretical

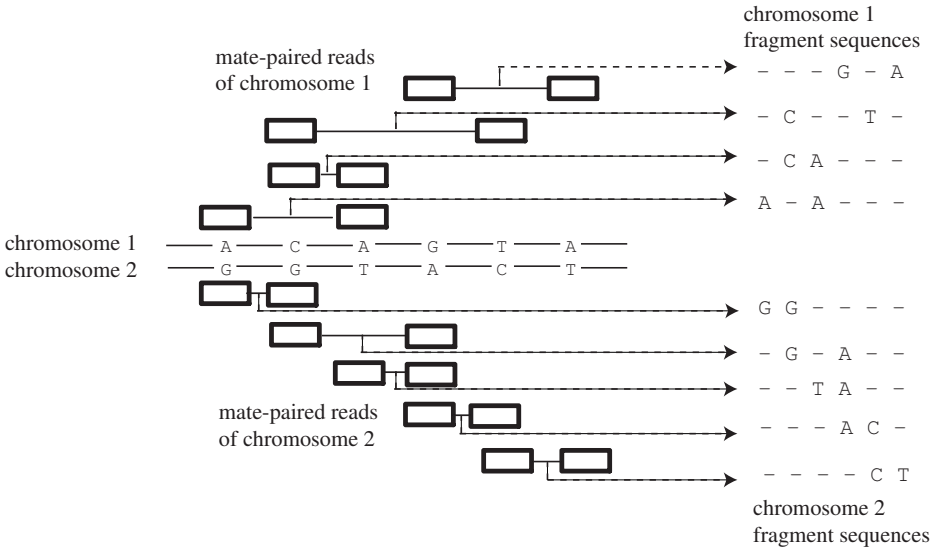

FIG. 1. *Mate-paired fragments from a diploid sequence. A pair of chromosome sequences containing a set of variable sites (center) are shotgun sequenced to produce a set of mate-paired fragments. Fragments occur at random positions along the genome, generally in the form of mate pairs corresponding to paired ends of individual cloned segments of DNA. Boxes linked by lines above and below the strands show the regions of sequence covered by the paired ends of each mate pair. Each mate pair will then cover some possibly discontiguous set of variable sites, which we can encode in a string (right). For each variable site, the string may contain the base (allele) value at the site or a dash for a site not covered by the mate pair. The collection of such strings forms the input to a haplotype assembly problem.*

results on their tractability [14, 19, 22, 1], and developed an assortment of exact [14, 19, 22, 1, 17, 34] and heuristic [20, 29, 31, 15, 32, 4] methods for solving variants of the problem. Aspects of this topic have previously been reviewed by Halldórsson et al. [10], Greenberg et al. [9], and Zhang et al. [35]. Haplotype assembly has since found new prominence with the inference of the first true haploid human genome sequence [15], which showed that haplotype assembly could produce large, accurate haplotypes on a genomic scale. With the advent of new high-throughput sequencing technologies, as well as more sophisticated statistical models of the haplotype assembly problem [16, 12, 13, 2], there is good reason to believe that haplotype assembly will become a standard part of whole-genome sequencing approaches.

This review surveys the history and possible future prospects of the haplotype assembly problem with a primary focus on computational theory and algorithms. We will first examine several formalizations of the problem as combinatorial optimizations. We will then explore a series of hardness results. We next cover algorithms for solving the problem in its various guises, which include both exact algorithms and heuristic approaches. We then discuss some statistical issues that have been raised in establishing long-range haplotype inferences and the degree of confidence one can

have in them. Finally, we discuss recent and possible future directions in haplotype assembly and related problems.

2. Theory and Problem Variants.

2.1. Notation and Problem Abstractions. To describe the haplotype assembly problem we must first develop a formal representation of the problem input. The input is presumed to be a set of haplotype fragments each covering some number of variant genomic sites, which we will assume here are SNPs. We will further assume that we have exactly two chromosomes and thus exactly two possible alleles at each variant site. We also assume that these fragments have been pre-aligned to a consensus genome assembly, meaning that we have a linear order of variant sites and that we know which fragments cover each site. As a result, we can represent our input as an $m \times n$ matrix F whose m rows correspond to m input fragments and whose n columns correspond to n SNP sites. We refer to row i of F as $\vec{f_i}$ and column j of row i as f_{ij}. Each element of F is drawn from the alphabet $\{0, 1, -\}$, where 0 and 1 refer by convention to the major (more frequent) and minor (less frequent) alleles at the site. '$-$' refers to a lack of information at a site, either because the fragment does not span the site in question or possibly because of a failure of a sequencing assay at a given site. We treat a mated pair of fragments as a single fragment, and hence single matrix row, with a gap corresponding to the unsequenced regions between the sequenced paired ends.

To understand the problem variants in the literature, we rely on a notion of agreement between fragments. Two fragments i and j are said to *conflict* if there exists some SNP site k such that

$$f_{ik} \neq f_{jk} \wedge f_{ik} \neq \text{`}-\text{'} \wedge f_{jk} \neq \text{`}-\text{'}$$

Informally, a conflict means that the fragments cover a common SNP site and have different values at that site. For error-free data, a conflict implies that the fragments come from an overlapping region of different chromosome copies. We can further define a distance between two fragments as the number of sites at which they conflict. If we assume n variant sites in total then

$$d(\vec{f_i}, \vec{f_j}) = \sum_{k=1}^{n} I(f_{ik} \neq f_{jk} \wedge f_{ik} \neq \text{`}-\text{'} \wedge f_{jk} \neq \text{`}-\text{'})$$

where $I(b)$ is an indicator function: $I(b) = 1$ for b true and $I(b) = 0$ for b false.

The goal of haplotype assembly is to partition the rows of F into two subsets F_1 and F_2 corresponding to the chromosomes. We do not necessarily require that $F_1 \cup F_2 = F$. Once we have F_1 and F_2, it is trivial to infer two consensus haplotype vectors

$$\vec{h_1} = \{h_{11}, h_{12}, \ldots, h_{1n}\}$$

$$\vec{h_2} = \{h_{21}, h_{22}, \ldots, h_{2n}\}$$

summarizing the partition F_1 and F_2, e.g., by taking the more common allele (if any) at each site covered by each part. The specifics of how we judge the quality of the partition $F_1 \cup F_2$ and form the consensus haplotypes $\vec{h_1}$ and $\vec{h_2}$ distinguish the different variants of the haplotype assembly problem.

In explaining the problem variants in the literature, it is helpful to consider two further abstractions of the data [14]. We first construct a *fragment conflict graph* $G_F = (V_F, E_F)$ where $|V_F| = m$ (i.e., one node per fragment) and the edge set is defined by the pairs of fragments that conflict:

$$E_F = \{(v_i, v_j) | v_i \in V_F, v_j \in V_F, d(\vec{f_i}, \vec{f_j}) > 0\}.$$

Fig. 2 illustrates a fragment conflict graph for a hypothetical data set. We can optionally treat G_F as a weighted graph, weighting each edge by the distance between its corresponding fragments:

$$w(v_i, v_j) = d(f_i, f_j).$$

If the data is error-free, then it must be possible to partition the fragments into two sets such that there are no conflicts within either set. Thus, error free data implies that G_F is bipartite; two fragments from the same chromosome will not conflict while two fragments from different chromosomes may or may not conflict. In Fig. 2, the grey line shows a partition of the fragments into two parts implying a solution to the fragment assignment problem. For bipartite G_f, the haplotype assembly problem trivially reduces to finding the two parts of a bipartite graph, which we can solve in linear time.

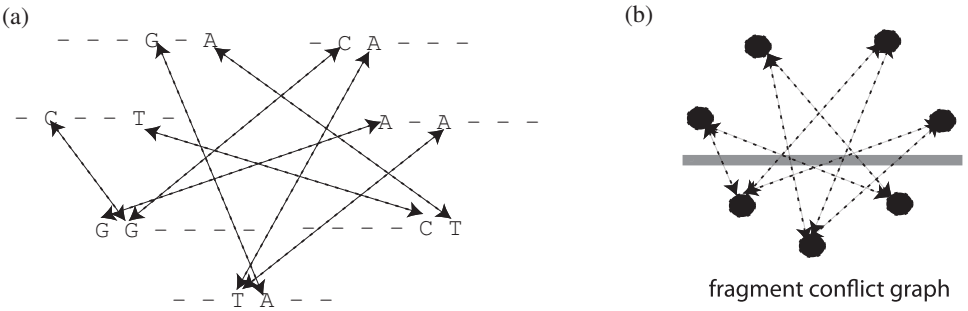

FIG. 2. *The fragment conflict graph. (a) A set of hypothetical fragments with edges labeling conflicts between pairs of fragments. (b) The fragment conflict graph G_F corresponding to the fragments in (a). The grey bar cuts all edges in the graph, showing that G_F is bipartite and that the haplotype assembly problem hence has an error-free solution.*

Some formulations of the haplotype assembly problem depend on an alternative graph formulation based on conflicts between SNP sites. For this formulation, we

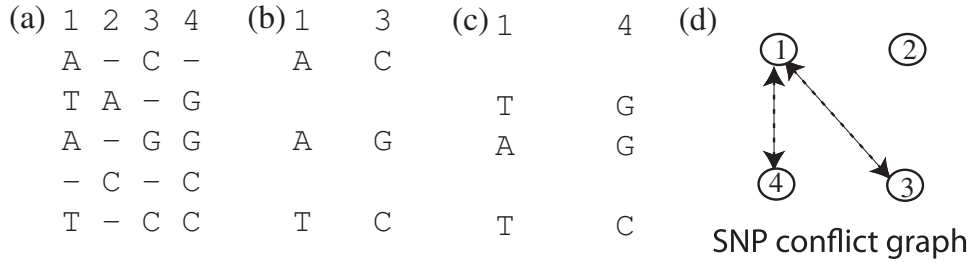

FIG. 3. *The SNP conflict graph. (a) A hypothetical data set consisting of five fragments sequenced at four sites. (b) Highlight of a SNP conflict between columns 1 and 3 of (a), using rows 1, 3, and 5. (c) Highlight of a SNP conflict between columns 1 and 4 of (a) using rows 2, 3, and 5. (d) The SNP conflict graph for (a), with edges corresponding to the conflicts shown in (b) and (c).*

define a *SNP conflict graph* $G_S = (V_S, E_S)$ specifying SNP pairs that are inconsistent with having come from at most two haplotypes. For the SNP conflict graph, $|V_S| = n$ (one vertex per SNP) and E_S is defined as follows:

$$E_S = \{(v_i, v_j) | v_i, v_j \in V_S \bigwedge$$
$\exists k_1, k_2, k_3$ s.t.
$((f_{k_1 i}, f_{k_1 j}) \neq (f_{k_2 i}, f_{k_2 j})) \wedge ((f_{k_1 i}, f_{k_1 j}) \neq (f_{k_3 i}, f_{k_3 j})) \wedge ((f_{k_2 i}, f_{k_2 j}) \neq (f_{k_3 i}, f_{k_3 j})) \wedge$
$(f_{k_1 i} \neq `-') \wedge (f_{k_1 j} \neq `-') \wedge (f_{k_2 i} \neq `-') \wedge (f_{k_2 j} \neq `-') \wedge (f_{k_3 i} \neq `-') \wedge (f_{k_3 j} \neq `-')\}$

Informally, E_S is the set of SNP pairs for which at least three haplotypes are observed across all genotypes. When two SNPs conflict then the set of fragments covering those sites cannot be resolved into two haplotypes without disagreements among the fragments of at least one haplotype. Fig. 3 illustrates the construction of G_S.

2.2. Problem Formulations. In general, data will not be error free and thus G_F may not be bipartite. Conflicts may occur because of sequencing errors, which introduce erroneous SNP values into individual fragments, and paralogous misrecruitment, which introduces erroneous fragments into the data set. Different problem formulations reflect different ways of inducing some bipartite G'_F close to G_F. Our first formulations of the haplotype assembly problem approximately captures the intuition that sequencing errors are the main source of conflicts in haplotypes:

> **Minimum edge removal (MER) [19]:** Find $V_1, V_2 \subseteq V_F$ such that $V_1 \cup V_2 = V_F$ minimizing $\sum_{v_i, v_j \in V_1} w(v_i, v_j) + \sum_{v_i, v_j \in V_2} w(v_i, v_j)$.

Note that although we define MER to minimize edge weights, we could alternatively define an unweighted version of the problem:

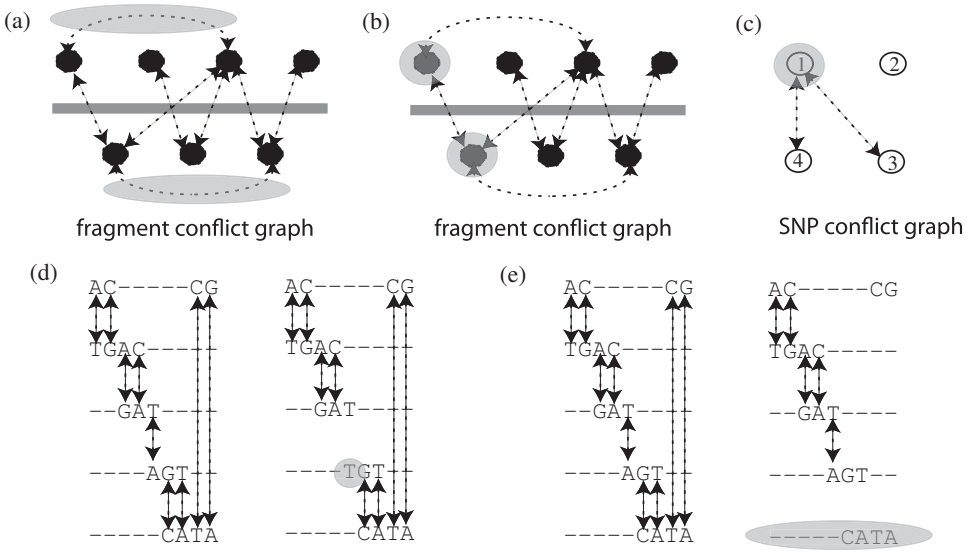

FIG. 4. *Illustration of haplotype assembly problem variants. (a) Minimum edge removal (MER). (b) Minimum fragment removal (MFR). (c) Minimum SNP removal (MSR). (d) Minimum error correction (MEC). (e) Longest fragment reconstruction (LFR).*

> **Unweighted minimum edge removal (UMER) [19]:** Find $V_1, V_2 \subseteq V_F$ such that $V_1 \cup V_2 = V_F$ minimizing $|\{(v_i, v_j) \in E_F | v_i, v_j \in V_1 \vee v_i, v_j \in V_2\}|$.

MER is illustrated in Fig. 4(a). We assume that we need to remove some subset of the edges of G_F (marked in grey in the figure) to produce a bipartite graph on all nodes. We may then have conflicting fragments assigned to a given chromosome and must choose consensus SNP alleles for each chromosome to derive the two haplotypes. Although it does not precisely correspond to any reasonable error model, MER can nonetheless be a useful formulation algorithmically because it reduces the the problem to a well-studied graph optimization problem, Maximum Cut [11].

An analogous formulation of the problem can also be derived from the fragment conflict graph:

> **Minimum fragment removal (MFR) [14]:** Find $V_1, V_2 \subseteq V_F$ minimizing $|V_F/(V_1 \cup V_2)|$ such that $V_1 \cap V_2 = \emptyset$ and $\nexists v_i, v_j \in V_1$ s.t. $(v_i, v_j) \in E$ and $\nexists v_i, v_j \in V_2$ s.t. $(v_i, v_j) \in E_F$.

Informally, the problem is to remove as few fragments as possible so as to leave a bipartite fragment conflict graph. The remaining graph will then be conflict-free and we can therefore easily derive the two haplotypes from its bipartition. This variant is illustrated in Fig. 4(b), which shows how we can remove some subset of the nodes of G_F (marked in grey) to produce a bipartite graph on the remaining nodes. MFR

intuitively corresponds to an error model in which we assume that conflicts arise because of extraneous fragments, as from paralogous misrecruitment, and that we can best solve the problem by finding a minimum set of extraneous fragments.

We can use the SNP conflict graph G_S to help us understand an alternative formulation of the problem:

> **Minimum SNP removal (MSR) [14]:** Find $V' \subseteq V_s$ minimizing $|V'|$ such that $\forall (u,v) \in E_s$ either $u \in V'$ or $v \in V'$.

Informally, MSR is the problem of locating a minimum number of SNP sites one needs to remove such that the full fragment set on the remaining sites can be assembled into two conflict-free haplotypes. The problem is equivalent to finding a maximum independent set in G_S. Lancia et al. [14] showed that the SNPs corresponding to any independent set in G_S induce a bipartite graph in G_F and hence correspond to an error-free solution to the problem. The MSR problem variant corresponds to the intuition that errors in data are likely to come from problematic sites on the genome where either a chance sequencing error or some unusual local sequence property leads to spurious conflicts. Fig. 4(c) illustrates the MSR problem, highlighting in grey a single SNP that can be removed from a hypothetical G_S to permit error-free haplotype assembly.

One especially popular variant in the literature that does not lend itself to a similarly simple graph construction is the following:

> **Minimum error correction (MEC) [19]:** Find a set of set of fragment/site tuples $X = \{(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)\}$ minimizing $|X| = k$ such that there is a bipartite fragment conflict graph G'_F induced by the matrix F', where $f'_{ij} = f_{ij}$ for $(i,j) \notin X$ and $f'_{ij} = \bar{f_{ij}}$ for $(i,j) \in X$.

For any SNP allele A, we define $\bar{A}$ to be the alternative allele at A's SNP site. MEC corresponds to a straightforward error-correction model of haplotype assembly: we want to flip as few sequenced sites as possible so as to allow the assembly of two error-free haplotypes. This problem is illustrated in Fig. 4(d), showing a hypothetical data set (left) that cannot be resolved into two haplotypes because it contains an odd-length conflict cycle and hence is not bipartite. By flipping one allele (right, highlighted in grey), we can break the cycle and produce a bipartite fragment set. This problem variant corresponds naturally to the intuition that spurious conflicts are introduced into the data through random sequencing errors and that the most plausible resolution is the one that requires assuming the fewest possible sequencing errors. Note that MEC is also sometimes called the **Minimum Letter Flip (MLF)** problem [9].

One final variant was proposed due to the practical desire to use haplotype as-

sembly to generate long haplotypes:

> **Longest haplotype reconstruction (LHR) [14]:** Find a subset $V' \subseteq V_f$ with induced edge set E' maximizing the set of SNP sites covered by any $v \in V'$ such that (V', E') is bipartite.

LHR identifies the longest conflict-free haploid region one can construct from a given fragment set. This may be a useful outcome when one is performing haplotype assembly to find candidate haplotypes for seeding diploid haplotype phasing algorithms. This problem is illustrated in Fig. 4(e), where the same incompatible fragment set as in Fig. 4(d) (left) is resolved by eliminating the bottom fragment (right, highlighted in grey). The result is a pair of haplotypes spanning the full length of the sequence. LHR does not correspond to any obviously reasonable error model for the data, however, and appears to have been abandoned in the subsequent literature. We will therefore devote minimal attention to it here.

In addition to these major variants, a few specialized formulations designed to deal with other data sources [36] and other variations on these basic techniques [18] have appeared in the literature and are discussed in the context of heuristic methods in Sec. 3.2 below.

3. Algorithms.

3.1. Exact Algorithms. Most of the problem variants examined here have been shown to be NP-hard, at least in the general case (arbitrary numbers and positions of gaps within fragments). Lancia et al. [14] showed MFR and MSR to be NP-hard in the general case. In fact, they showed MFR to be hard even if each fragment is limited to a single gap and MSR to be hard even if each fragment is limited to at most two gaps. Lippert et al. [19] have shown MER to be NP-hard in the general case. Zhao et al. [37] showed MEC and two weighted variants on that problem to be hard even in the case of no internal gaps in fragments. Rizzi et al. [22, 1] have further established both MFR and MSR to be APX-hard (not approximable to any constant bound in polynomial time) in the case of general gaps.

In some special cases, though, the haplotype assembly problems we examine are efficiently solvable. A degenerate version of the problem is to assume that the fragment matrix F is *gapless*, meaning that each fragment covers a contiguous set of sites on the genome:

$$\forall i, j, k, l \text{ such that } j < k < l,\ (f_{ij} \neq \text{`}-\text{'} \wedge f_{il} \neq \text{`}-\text{'}) \Longrightarrow f_{ik} \neq \text{`}-\text{'}$$

Lancia et al. [14] showed that for all three variants they proposed (MFR, MSR, and LHR), optimal inference can be performed in polynomial time in the gapless case. While these algorithms were used as proofs of polynomial run-time, they were not practical for realistic data sets. Rizzi et al. [22, 1] later improved on these algorithms

to develop the first practical algorithms for gapless MFR and gapless MSR, using dynamic programming to provide run times of $O(m^2n + m^3)$ for MFR and $O(mn^2)$ for MSR. Li et al. [17] extended this tractability result to the k-MFR problem, a generalization of MFR allowing for k haplotypes (standard MFR is then 2-MFR). They constructed an $O(m^2n + m^{k+1})$ algorithm for gapless k-MFR. As noted above, MEC remains hard even in the gapless case [37].

Several methods have also been developed to allow practical solution of instances of the hard versions of the problem. One popular method for solving hard problems of this sort in practice is branch-and-bound. Lippert et al. [19] developed the first practical approach for gapped haplotype assembly through a branch-and-bound algorithm for MFR. They described a basic variant of this algorithm and also suggested some more sophisticated cut heuristics that may prove helpful on especially hard instances. Wang et al. [30] developed an alternative branch-and-bound approach to the MEC problem.

An alternative approach has been to develop fixed parameter tractable (FPT) algorithms. Rizzi et al. [22] showed the MFR and MSR variants to be fixed parameter tractable in a parameter k corresponding to the maximum number of gap positions per fragment between the first and last non-gap sites. Both methods use dynamic programming over possible haplotypes within windows covering these gap segments to achieve run-times exponential in k but otherwise polynomial in problem size. The resulting methods allow run times of $O(2^{2k}nm^2 + 2^{3k}m^3)$ for MFR and $O(mn^{2k+2})$ for MSR. Xie and Wang [34] noted that these approaches can be impractical for mate-paired fragments because k can be quite large due to the number of gap sites between paired ends. They therefore proposed a more involved FPT algorithm for MFR parameterized with two new parameters — k_1, the maximum number of SNP sites covered by any fragment, and k_2, the maximum number of fragments covering any SNP site — and showed a run time of $O(nk_2 3^{k_2} + m \log m + mk_1)$. Because k_1 is at most n and k_2 is generally below 10, this method should be far more practical for mate-paired data. Li et al. [17] developed an FPT approach for k-MFR, yielding run-time $O(2^{2\kappa}m^2n + 2^{(k+1)\kappa}m^{\kappa+1})$, where κ is equivalent to Rizzi et al.'s k above.

A third common approach to solving similar hard combinatorial problems is the use of integer linear programming (ILP) reductions. Lippert et al. [19] reported having attempted such an approach that did not perform adequately in practice. They provide no details on the method, though. Li et al. [17] describe an ILP reduction for k-MFR, showing that it can be posed as a simple ILP and therefore is in principle amenable to methods for ILP solution. The ILP strategy has thus so far received comparatively little attention in the haplotype assembly field.

3.2. Optimization Heuristics.

Several heuristics were developed based on stochastic models of the process of

fragment generation. One approach devised by Wang et al. [31] treats recruitment of fragments to haplotypes as a Markov process in which successive allele pairs are generated based on an examination of the preceding d SNP sites in the sequence, where d is some small constant (often 1). Given this model, a probability for each possible phasing of a given site can be estimated from the frequencies with which alleles are shared in the fragments spanning the current site and those immediately preceding it. A Viterbi-like dynamic programming algorithm can then find the most probable haplotype pair for the fragment set. Chen et al. [4] developed a linear-time greedy recruitment algorithm and two randomized variants of that algorithm for a probabilistic variant of the MEC model, assuming conflicts arise from uniformly random sequencing errors. They showed that under some assumptions about sequencing errors in the data it was possible to prove a series of rigorous error bounds for these fast, simple methods. Randomized approaches have also been used for pure optimization versions of the problem. For example, Wang et al. [30] developed a randomized genetic algorithm, concatenating the prefix of one solution to a suffix of another as their mating operation and flipping random positions as their mutation operation.

Vinson et al. [29] developed their own novel heuristic approach for a practical project sequencing the highly polymorphic sea squirt (*Ciona savignyi*) genome by building haplotype assembly directly into the normal genome assembly process. High polymorphism is normally problematic for a genome assembly, but it is an advantage in haplotype assembly because it increases the average amount of phase information available per fragment. Rather than inferring a consensus genome and then using it to align fragments, they heuristically introduced a penalty function to the Arachne genome assembler [3] to determine whether two imperfectly matched fragments might correspond to the same genome location. The result was to force the assembler to construct distinct assemblies for each haplotype, which could then be aligned after that fact. While this approach does not solve for a precise specification of the problem, it has the advantage of circumventing problems genomic polymorphism might otherwise introduce at the assembly stage.

Lindsay et al. [18] developed a greedy algorithm that solves for a novel weight function similar to the MER objective. Their approach repeatedly merges sets of fragments until only two sets are left, greedily choosing the two overlapping sets with minimal conflict. A similar greedy approach was chosen for the FastHare heuristic method [20], which scans linearly across the genome greedily growing a haplotype pair. This approach is meant to solve both for accuracy of haplotype reconstruction and for an additional problem variant, Minimum Element Removal (MER), corresponding to minimizing the number of fragment SNPs one must convert to gaps in order to construct two error-free haplotypes.

The largest haplotype assembly project to date, that of Levy et al. [15] in assembling the first diploid human genome, adopted a similar greedy approach with some

refinements. This heuristic approach began by arbitrarily assigning one fragment to a haplotype and then repeatedly identified the fragment with the most overlap with the existing assignment of heterozygous alleles to merge with one of the two growing haplotypes. The process would continue until no more fragments could be recruited to the current haplotype pair and then start anew at a different location on the genome. Finally, they applied a series of greedy refinements consisting of re-estimating the haplotypes at a single site based on a majority vote of the assigned fragments and reassigning individual fragments based on a majority vote of the fragment's covered SNP sites. These assignments were subsequently fed to an MCMC refinement method, discussed in Sec. 3.3. Their study showed surprising success in assembling large, accurate haplotypes from 7.5X of sequence coverage. They report that half of their variant sites were assembled into haplotypes of more than 400 variants. Collectively, approximately half the genome was covered with haplotypes of over 200 kb each. Further, the haplotypes found were 97.4% consistent with the individual fragment data and showed high correspondence to high-LD haplotype pairs found in the HapMap [25].

Some work has attempted to improve haplotype quality by introducing the assumption of additional genotype data, which is easier to gather than fragment data and may reduce the solution space. Zhang et al. [36] introduced this method through a variant of MEC that they call Minimum Conflict Individual Haplotyping (MCIH), which consists of optimizing for the MEC variant but subject to the additional constraint that the inferred haplotypes must correspond to possible phasings of a separate unphased genotype data set. They developed a dynamic programming algorithm to solve the MCIH problem in time exponential in the maximum fragment length (excluding leading and ending gaps) but otherwise polynomial in problem size. They further derived a neural network approach for classifying fragments into two haplotypes for the same problem variant. Wang et al. [32] proposed a similar model of haplotype assembly augmented with external genotype information and developed a greedy hill-climbing search heuristic for that model.

3.3. Sampling Methods. One issue largely neglected in the preceding formalizations is the uncertainty involved in haplotype estimation. A simple version of uncertainty arises from the fact that incomplete data will usually produce many optimal solutions to any given formalization of the problem. This multiplicity of solutions occurs primarily because two fragments can only be phased relative to one another if there is a chain of pairwise fragment conflicts linking them, corresponding to a path in the fragment conflict graph. Distinct components of the fragment conflict graph will thus need to be phased arbitrarily relative to one another, creating in general 2^{k-1} optimal solutions for a k-component conflict graph. Even over connected regions, though, there may be some uncertainty due to the possibility of erroneous data. If we consider the possibility that some conflicts may in fact be spurious then all phase as-

signments will have some probability of error, with the probability of correct phasing generally decreasing with increasing assembly length.

Li et al. [16] developed the first approach to place this haplotype inference problem on a more rigorous probabilistic footing. Their methods built on the Churchill and Waterman [5] theory for analyzing uncertainty in assembling a single genome. The theory established a probabilistic model in which there is presumed to be some probability that any given variant site is incorrectly sequenced. Any two overlapping fragment pairs will therefore have some probability of being mistakenly inferred to be in conflict when they in fact come from the same haplotype, or to be mistakenly inferred not to be in conflict when they in fact come from opposite haplotypes that disagree within their region of overlap. This basic model allows for the derivation of confidence bounds on any haplotype reconstruction based on accumulated potential errors across the fragment assignments. Li et al. proposed a Gibbs sampler to sample over possible partitions of fragments within their probabilistic framework, making possible the discovery of high-confidence regions of haplotype assignments and the derivation of confidence bounds on those assignments.

This approach was extended by Kim et al. [12, 13] for sequencing the *Ciona intestinalis* genome, a relative of the *Ciona savignyi* organism assembled by Vinson et al. [29] that has a similarly high polymorphism rate. Kim et al. [12] developed an efficient method for calculating likelihoods using a somewhat generalized version of the model of Li et al. and showed how these likelihood calculations could be used to improve construction of high-confidence haplotypes. Kim et al. [13] further improved on the method by extending their probability model to incorporate some additional problem-specific data, such as confidence scores on individual base calls during sequencing. They also extended the Gibbs sampler of Li et al. [16] to start with local haplotypes built from pairs of SNP sites and incorporate sampling transitions to successively merge adjacent haplotypes as well as rephase within these local haplotypes. This bottom-up process continues until estimated confidences become too small to support further extension. This approach allowed for the more efficient construction of large, high-confidence haplotypes.

Levy et al. [15] added a similar sampling step subsequent to their greedy haplotype assembly strategy of the human genome. They report that this MCMC sampling reduced conflicts between their assembly and known HapMap [25] by 30% relative to their initial assembly. A similar sampling approach was adopted by Bansal et al [2]. They developed a Metropolis sampler to enumerate over possible haplotypes for which the move set consists of flipping the phase of some subset of the possible haplotypes. To find good candidate cuts, they rely on a hierarchical partitioning of the SNP conflict graph constructed by searching for low-weight graph cuts. These low-cost cuts identify points of high uncertainty in the phase assignments, which allows far more efficient sampling of the haplotype space than do more naïve approaches

to MCMC sampling. Bansal et al. reanalyzed Levy et al.'s data with their method, showing that the haplotypes produced by their method could cover half of the genome with haplotypes over 350 kb. They further estimated, from comparison to HapMap haplotypes, an error of under 1% in phasing adjacent sites in the diploid human genome.

4. Discussion. The haplotype assembly problem has developed from initially a specialized and largely theoretical curiosity to a valuable component of the sequencing of diploid genomes. In the process, the problem has undergone a variety of transformations from simple combinatorial variants that lent themselves well to theoretical analysis to sophisticated probabilistic models that allow examination of complex data sources and estimation of rigorous confidence bounds on assembly. The most advanced methods can now accurately cover a majority of the human genome with haplotypes of several hundred kilobases using levels of sequence coverage typical of conventional shotgun assembly studies.

The question remains what future there may be for computational work on this problem. The substantial advantages offered by the statistical approaches would seem to suggest that they are the future for haplotype assembly. Nonetheless, as Bansal et al. [2] show, sophisticated optimization can be a key component of building an efficient sampling approach. There is likely substantial additional work to be done on developing more realistic error models and developing algorithms for sampling efficiently from those models. Merging haplotype assembly with the normal assembly process, as was done by Vinson et al. [29] would also appear to be a natural direction for the field. There, too, there would seem to be a great deal of room for improvement of probabilistic models of haploid assembly in the presence of polymorphism and for genuinely new assembly algorithms to handle this form of assembly more efficiently. There may also be important practical work still to be done to handle harder problem variants than those considered here. The k-MFR problem [17] points to one such hard problem that may prove valuable in practice: multiplex assembly of more than two haplotypes. For example, such an approach may prove valuable for working with data derived from environmental sequencing projects where contributions are likely to come from many members of a given species as well as many species. As with the haplotype assembly problem to date, biological research practice can be expected to continue drive the work in exciting and unanticipated directions.

Acknowledgments. The author is grateful to S. Istrail for advice on the preparation of this manuscript. R.S. is supported by NSF IIS Grant No. 0612099.

REFERENCES

[1] V. Bafna, S. Istrail, G. Lancia, and R. Rizzi. *Polynomial and APX-hard cases of the individual haplotyping problem.* Theoretical Computer Science, 335(2005), pp. 109–125.

[2] V. Bansal, A. L. Halpern, N. Axelrod, and V. Bafna. *An MCMC algorithm for haplotype assembly from whole-genome sequence data*. Genome Research, 18(2008), pp. 1336–1346.

[3] S. Batzoglou, D. B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J. P. Mesirov, and E. S. Lander. *ARACHNE: A whole-genome shotgun assembler*. Genome Research, 12(2002), pp. 177–189.

[4] Z. Chen, B. Fu, R. Schweller, B. Yang, Z. Zhao, and B. Zhu. *Linear time probabilistic algorithms for the singular haplotype reconstruction problem from SNP fragments*. Journal of Computational Biology, 15(2008), pp. 535–546.

[5] G. A. Churchill and M. S. Waterman. *The accuracy of DNA sequences: Estimating sequence quality*. Genomics, 14(1992), 89–98.

[6] A. G. Clark. *Inference of haplotypes from PCR-amplified samples of diploid populations*. Molecular Biology and Evolution, 7(1990), pp. 111–122.

[7] A. Edwards, H. Voss, P. Rice, A. Civitello, J. Stegemann, C. Schwager, J. Zimmerman, H. Erfle, C. T. Caskey, and W. Ansorge. *Automated DNA sequencing of the human HPRT locus*. Genomics, 6(1990), pp. 593–608.

[8] R. .D. Fleischmann, M. D. Adams, and O. White et al. *Whole-genome random sequencing and assembly of Haemophilus influenza Rd*. Science, 269(1995), pp. 496–512.

[9] H. J. Greenberg, W. E. Hart, and G. Lancia. *Opportunities for combinatorial optimization in computational biology*. INFORMS Journal on Computing, 16(2004), pp. 211–231.

[10] B. Halldórsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. *Combinatorial problems arising in SNP and haplotype analysis*. In: S. Istrail, editor, SNPs and Haplotype Inferences (LNCS 2731), pages 26–47, 2004.

[11] R. M. Karp. *Reducibility among combinatorial problems*, pages 85–90. Plenum Press, 1972.

[12] J. H. Kim, M. S. Waterman, and L. M. Li. *Accuracy assessment of diploid consensus sequences*. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 4(2007), pp. 88–97.

[13] J. H. Kim, M. S. Waterman, and L. M. Li. *Diploid genome reconstruction of ciona intestinalis and comparative analysis with ciona savignyi*. Genome Research, 17(2008), pp. 1101–1110.

[14] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz. *SNPs problems, complexity, and algorithms*. In: F. Meyer auf der Heide, editor, European Symposium on Algorithms (LNCS 2161), pages 182–193, 2001.

[15] S. Levy, G. Sutton, and P. C. Ng et al. *The diploid genome sequence of an individual human*. PLoS Biology, 5(2007), pp. 2113–2144.

[16] L. M. Li, J. H. Kim, and M. S. Waterman. *Haplotype reconstruction from SNP alignment*. Journal of Computational Biology, 11(2004), pp. 507–518.

[17] Z.-P. Li, L.-Y. Wu, Y.-Y. Zhao, and X.-S. Zhang. *A dynamic programming algorithm for the k-haplotyping problem*. Acta Mathematicae Applicatae Sinica, 22(2006), pp. 405–412.

[18] S. J. Lindsay, J. K. Bonfield, and M. E. Hurles. *Shotgun haplotyping: a novel method for surveying allelic sequence variation*. Nucleic Acids Research, 33(2005), pp. e152.

[19] R. Lippert, R. Schwartz, G. Lancia, and S. Istrail. *Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem*. Briefings in Bioinformatics, 3(2002), pp. 23–31.

[20] A. Panconesi and M. Sozio. *Fast Hare: A fast heuristic for single individual SNP haplotype reconstruction*. In: Workshop on Algorithms in Bioinformatics (LNBI 3240), pages 266–277, 2004.

[21] R. Redon, S. Ishikawa, and K. R. Fitch et al. *Global variation in copy number in the human genome*. Nature, 444(2006), pp. 444–454.

[22] R. Rizzi, V. Bafna, S. Istrail, and G. Lancia. *Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem*. In: R. Guigó and D. Gusfield, editors, Workshop on Algorithms in Bioinformatics (LNCS 2452), pages 29–43, 2002.

[23] S. T. Sherry, M.-H. Ward, and K. Sirotkin. *dbSNP — database for single nucleotide polymorphisms and other classes of minor genetic variations.* Genome Research, 9(1999), pp. 677–679.

[24] The International HapMap Consortium. *The international HapMap project.* Nature, 426(2005), pp. 789–796.

[25] The International HapMap Consortium. *A second generation human haplotype map of over 3.1 million SNPs.* Nature, 449(2007), pp. 851–861.

[26] The International Human Genome Sequencing Consortium. *Initial sequencing and analysis of the human genome.* Nature, 304(2001), pp. 412–417.

[27] E. Tunzun, A. J. Sharp, J. A. Bailey, R. Kaul, V. A. Morrison, L. M. Pertz, E. Haugen, H. Hayden, D. Albertson, D. Pinkel, M. V. Olson, and E. E. Eichler. *Fine-scale structural variation of the human genome.* Nature Genetics, 37(2005), pp. 727–732.

[28] J. C. Venter, M. D. Adams, and E. W. Myers et al. *The sequence of the human genome.* Science, 291(2001), pp. 1145–1434.

[29] J. P. Vinson, D. B. Jaffe, K. O'Neill, E. K. Karlsson, N. Strange-Thomann, S. Anderson, J. P. Mesirov, N. Satoh, Y. Satou, C. Nusbaum, B. Birren, J. E. Galagan, and E. S. Lander. *Assembly of polymorphic genomes: Algorithms and application to Ciona savignyi.* Genome Research, 15(2005), pp. 1127–1135.

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

[31] R.-S. Wang, L.-Y. Wu, X.-S. Zhang, and L. Chen. *A Markov chain model for haplotype assembly from SNP fragments.* Genome Informatics, 17(2006), pp. 162–171.

[32] Y. Wang, E. Feng, R. Wang, and D. Zhang. *The haplotype assembly model with genotype information and iterative local-exhaustive search algorithm.* Computational Biology and Chemistry, 31(1007), pp. 288–293.

[33] J. L. Weber and E. W. Myers. *Whole-genome shotgun sequencing.* Genome Research, 7(197), pp. 401–409.

[34] M. Xie and J. Wang. *An improved (and practical) parameterized algorithm for the individual haplotyping problem MFR with mate-pairs.* Algorithmica, 52(2008), pp. 250–266.

[35] X.-S. Zhang, R.-S. Wang, L.-Y. Wu, and L. Chen. *Models and algorithms for haplotyping problems.* Current Bioinformatics, 1(2006), pp. 105–114.

[36] X.-S. Zhang, R.-S. Wang, L.-Y. Wu, and W. Zhang. *Minimum conflict individual haplotype from SNP fragments and related genotype.* Evolutionary Bioinformatics, 2(2006), pp. 2456–2462.

[37] Y.-Y. Zhao, L.-Y. Wu, J.-H. Zhang, R.-S. Wang, and X.-S. Zhang. *Haplotype assembly from aligned weighted SNP fragments.* Computational Biology and Chemistry, 29(2005), pp. 281–287.