Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem

Ross Lippert, Russell Schwartz, Giuseppe Lancia and Sorin Istrail

Received (in revised form): 15th November 2001

Ross Lippert
is a member of the Informatics Research department at Celera Genomics in the USA. His research interests currently include exact string matching, haplotype inference, and high performance computation.

Russell Schwartz
is a member of the Informatics Research department at Celera Genomics. His research interests include simulation of biochemical systems, sequence analysis and data mining, and computational methods for studying genome polymorphisms.

Giuseppe Lancia
is currently an assistant professor in Operations Research at the University of Padova. His interests include algorithms for protein structure alignment, the combinatorics of genome comparison and combinatorial optimization in general.

Sorin Istrail
is a Senior Director of Informatics Research at Celera Genomics. His research interests include computational molecular biology, algorithms and computational complexity, and computational statistical mechanics.

Keywords: *polymorphism, genetics, genomics, algorithm, complexity*

Ross Lippert and Sorin Istrail,
Informatics Research,
Celera Genomics,
45 West Gude Drive,
Rockville,
MD 20850, USA

E-mail: Ross.Lippert@celera.com
Sorin.Istrail@celera.com

Abstract

With the consensus human genome sequenced and many other sequencing projects at varying stages of completion, greater attention is being paid to the genetic differences among individuals and the abilities of those differences to predict phenotypes. A significant obstacle to such work is the difficulty and expense of determining haplotypes – sets of variants genetically linked because of their proximity on the genome – for large numbers of individuals for use in association studies. This paper presents some algorithmic considerations in a new approach for haplotype determination: inferring haplotypes from localised polymorphism data gathered from short genome 'fragments.' Formalised models of the biological system under consideration are examined, given a variety of assumptions about the goal of the problem and the character of optimal solutions. Some theoretical results and algorithms for handling haplotype assembly given the different models are then sketched. The primary conclusion is that some important simplified variants of the problem yield tractable problems while more general variants tend to be intractable in the worst case.

INTRODUCTION

With complete genome sequences now available for humans[1,2] as well as many other important organisms, a major challenge for the field will be applying genomic data to locate genetic variants, or polymorphisms, that are predictive of disease. Of particular interest have been the single nucleotide polymorphisms (SNPs) at which a single DNA base varies from one individual to another. SNP maps for use in association studies have been generated by Celera[1] and by public sequencing efforts.[3–5] Individual SNPs can themselves be combined into haplotypes, sets of polymorphisms in a region that tend to be inherited together because of their proximity on the genome. Recent work indicates that haplotypes generally have more information content than individual SNPs,[6] but they are substantially more difficult to determine than genotypes or individual SNPs.

In practice, there are several methods for determining haplotypes, each with some strengths and weaknesses. Classically, pedigree information has been used to infer probable haplotypes,[7,8] providing an inexpensive but inaccurate method. In addition, sequencing of clones of a region of interest can directly determine haplotypes,[9,10] an accurate but slow and expensive process. Ruano *et al.*[11] introduced a direct experimental method to determine haplotypes through dilution of DNA samples to single molecules, followed by polymerase chain reaction (PCR) amplification, greatly reducing the cost of producing accurate haplotypes. Computational methods have sought to further reduce the cost of determining haplotypes of many individuals by inferring probable haplotypes given more easily obtained genotype information. Parsimony methods[12,13] formulate the problem as a combinatorial optimisation in which a program attempts to explain a data set with as few haplotypes as possible,

handling large haplotypes and large data sets but potentially giving incomplete results. Statistical algorithms[14–18] use inexact methods to find solutions to more complicated statistical models. These statistical methods give data that is generally more complete than that offered by parsimony methods, but potentially less accurate.

This paper discusses algorithmic issues in applying an alternative method for haplotyping based on the data and methodology of shotgun sequence assembly.[19,20] Specifically, we examine how haplotypes over long regions can be determined from short (single-stranded) sequence fragments, each covering only a few polymorphic sites. Figure 1 illustrates the general idea of how short fragments aligned to a genome sequence can be partitioned into two sets using conflicting SNP values between fragments, with the two sets determining the distinct haplotypes. The input data can come directly from a shotgun sequencing project (post-assembly) or might be generated specifically by a resequencing effort for the purpose of large-scale haplotyping. Haplotypes assembled by our method might be used for validation of a fragment assembly, as 'seed data' to improve the performance of other computational methods, or directly in disease association studies. This paper first presents some theoretical models of this 'SNP haplotype assembly problem' under different assumptions and objectives. We then describe some results on the hardness of the problems and methods for tractable variants. Finally, we discuss directions for future progress on this problem.

We seek to partition fragments in an assembly according to SNP allele values to recover long range haplotypes

SNP ASSEMBLY PROBLEMS

A SNP assembly is a tuple $(\mathscr{S}, \mathscr{F}, \mathscr{R})$ that consists of a set $\mathscr{S}$ of n SNPs, a set $\mathscr{F}$ of m fragments, and a covering relation $\mathscr{R}$: $\mathscr{S} \times \mathscr{F} \rightarrow \{O, A, B\}$ indicating whether s occurs on f

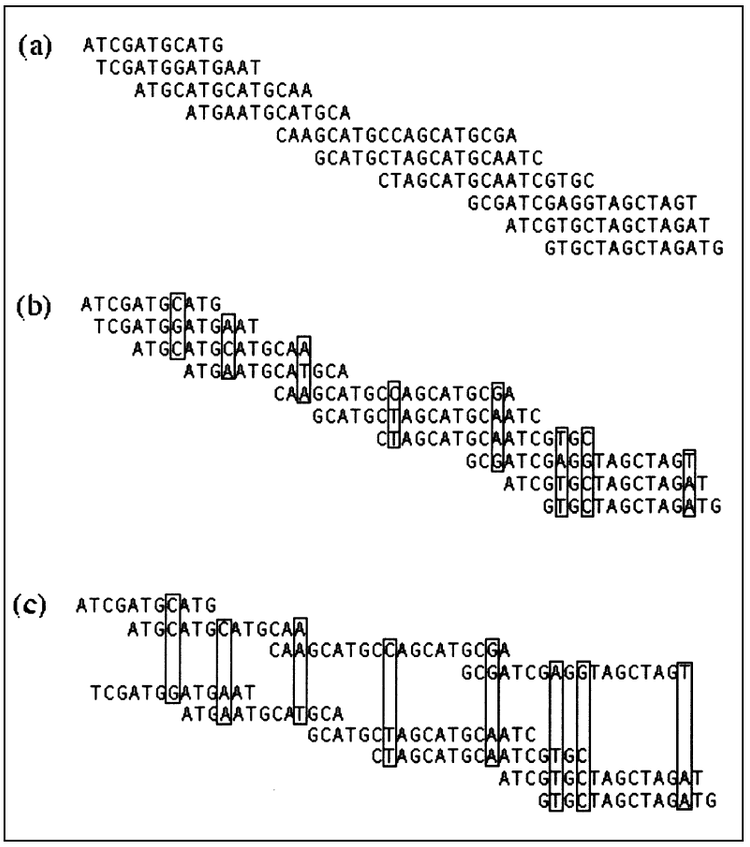

Figure 1: An illustration of partitioning fragments by SNP alleles: (a) a set of hypothetical fragments aligned according to overlaps; (b) the fragments with SNP loci marked; (c) the fragments set partitioned into two haplotypes according to SNP allele values

In the presence of errors we seek the 'most consistent' partition

(denoted with O) and, if occurring, with which flavour of s (A or B) occurs. The set of *proper SNPs* is $\mathcal{S}^* = \{s \in \mathcal{S} | \exists f, f' \in \mathcal{F}: \mathcal{R}(s, f) = A \wedge \mathcal{R}(s, f') = B\}$, and the set of *proper fragments* is $\mathcal{F}^* = \{f \in \mathcal{F} | \exists s \in \mathcal{S}: \mathcal{R}(s, f) \neq O\}$.

Assuming an ordering on $\mathcal{S} = \{s_i\}_{i=1\ldots n}$ and $\mathcal{F} = \{f_i\}_{i=1\ldots m}$, the *SNP matrix* is the $m \times n$ matrix over $\{O, A, B\}$ with values $\mathcal{R}(s_j, f_i)$ at row i column j.

Various objective functions can be used to define consistency. The problem is then a combinatorial optimisation.

Although not always the case, in some important applications (eg SNP assemblies arising from gapless fragments such as expressed sequence tags (ESTs), $\mathcal{R}$ has a *consecutive 1s property* (or C1P), where there exists an ordering of $\mathcal{S}$ (called a *consecutive 1s ordering*) such that for each row of the SNP matrix, the non-O values are all consecutive.

A *haplotyping* is a partition of $\mathcal{F}$ into two blocks H_1, H_2 called *haplotypes*. We say that an SNP assembly is *feasible* when there exists a haplotyping such that $\forall s \in \mathcal{S}, \forall f, f' \in H_i$: $(\mathcal{R}(s, f) = \mathcal{R}(s, f')) \vee (\mathcal{R}(s, f) = O) \vee (\mathcal{R}(s, f') = O)$, for $i = 1, 2$.

A fundamental problem in SNP analysis is, for any SNP assembly, $\mathcal{A} = (\mathcal{S}, \mathcal{F}, \mathcal{R})$, to find a *nearby* assembly $\bar{\mathcal{A}} = (\overline{\mathcal{S}}, \overline{\mathcal{F}}, \overline{\mathcal{R}})$ that is feasible and a minimum of some objective function $\text{obj}(\mathcal{A}, \bar{\mathcal{A}})$. There are a variety of optimisation criteria that one might apply to SNP assembly problems, eg:

With additional assumptions on the fragments, some of these optimisations are tractable

- MFR (minimum fragment removal): $\bar{\mathcal{A}} = (\mathcal{S}, \overline{\mathcal{F}}, \mathcal{R}|_{\mathcal{S} \times \overline{\mathcal{F}}})$, where $\overline{\mathcal{F}} \subset \mathcal{F}$ and $\text{obj}(\mathcal{A}, \bar{\mathcal{A}} = |\mathcal{F} - \overline{\mathcal{F}}|$.
- MSR (minimum SNP removal): $\bar{\mathcal{A}} = (\overline{\mathcal{S}}, \mathcal{F}, \mathcal{R}|_{\overline{\mathcal{S}} \times \mathcal{F}})$, where $\overline{\mathcal{S}} \subset \mathcal{S}$ and $\text{obj}(\mathcal{A}, \bar{\mathcal{A}} = |\mathcal{S} - \overline{\mathcal{S}}|$.
- MISR (minimum implicit SNP removal): $\bar{\mathcal{A}} = (\mathcal{S}, \overline{\mathcal{F}}, \mathcal{R}|_{\mathcal{S} \times \overline{\mathcal{F}}})$, where $\overline{\mathcal{F}} \subset \mathcal{F}$ and $\text{obj}(\mathcal{A}, \bar{\mathcal{A}} = -|\mathcal{S}^*|$ calculated with respect to $\bar{\mathcal{A}}$.
- MIFR (minimum implicit fragment removal): $\bar{\mathcal{A}} = (\overline{\mathcal{S}}, \mathcal{F}, \mathcal{R}|_{\overline{\mathcal{S}} \times \mathcal{F}})$, where $\overline{\mathcal{S}} \subset \mathcal{S}$ and $obj(\mathcal{A}, \bar{\mathcal{A}} = -|\mathcal{F}^*|$.
- MEC (minimum error correction): $\bar{\mathcal{A}} = (\mathcal{S}, \mathcal{F}, \overline{\mathcal{R}})$, where $\text{obj}(\mathcal{A}, \bar{\mathcal{A}} = |\{s \in \mathcal{S}, f \in \mathcal{F}: \mathcal{R}(s, f) \neq \overline{\mathcal{R}}(s, f)\}|$.

We will return to some of these problems in later sections.

Conflict graphs

We define the *fragment conflict graph*, $G_{\mathcal{F}} = (V, E)$ with nodes $V = \mathcal{F}$ and edge set $E = \{(f, f') | \exists s \in \mathcal{S}, (\mathcal{R}(s, f) \neq O) \wedge (\mathcal{R}(s, f') \neq O) \wedge (\mathcal{R}(s, f) \neq \mathcal{R}(s, f'))\}$. If the SNP problem is feasible, then $G_{\mathcal{F}}$ is bipartite, since a given haplotyping H_1, H_2 defines the shores of $G_{\mathcal{F}}$. Conversely, if $G_{\mathcal{F}}$ is bipartite with shores H_1 and H_2, then H_1, H_2 can be taken as a partition of $\mathcal{F}$ defining a haplotyping, and thus the SNP assembly is feasible. Figure 2 shows an example.

We define the *SNP conflict graph*, $G_{\mathcal{S}} = (V, E)$ with nodes $V = \mathcal{S}$ and edge set $E \subset \mathcal{S} \times \mathcal{S}$ defined by $(s_1, s_2) \in E$ iff $s_1, s_2 \in \mathcal{S}^*$ and there exists $f_1, f_2 \in \mathcal{F}$ such that the four element multiset $\{\mathcal{R}(s_i, f_i)\}$ consists of 3 As and 1 B or 3 Bs and 1 A.

Theorem 1 *Lancia When the SNP matrix has the consecutive 1s property, then the following hold:*

- *$G_{\mathcal{S}}$ is a perfect graph.*
- *$G_{\mathcal{F}}$ is bipartite iff $G_{\mathcal{S}}$ is an independent set.*

THEORY AND ALGORITHMS

We restate here the theory given by Lancia *et al.*[21] relating to the computational complexity of the various feasibility problems described in the previous section.

In general, they are NP-hard.

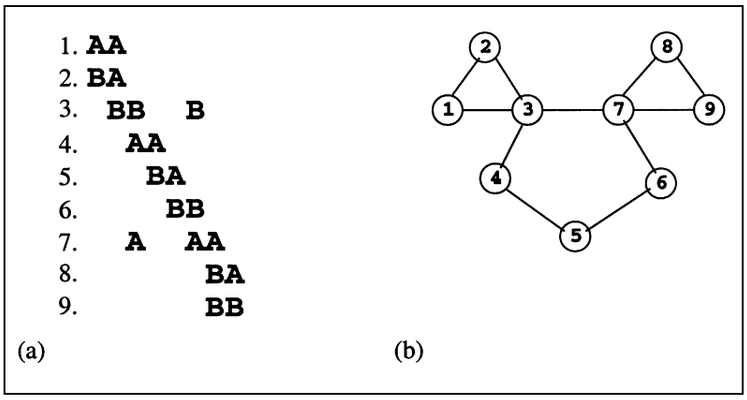

Figure 2: An example SNP matrix, with fragments 1 to 9, and its associated fragment conflict graph (Os omitted)

Generally these optimisations are formally intractable

Theorem 2 *For a general SNP matrix:*

- *MFR equivalent to* MAX Induced Bipartite Subgraph
- *MSR, MEC reduce from* MAXCUT

which makes them intractable.

The other NP-hardness results remain open.

The theorem relies on the following lemma.

Lemma 1 *Conflict graph generality*
Let $G = (V, E)$. *Then there exists a SNP assembly,* $\mathcal{A} = (\mathcal{S}, \mathcal{F}, \mathcal{R})$, *such that* $G_{\mathcal{F}} = G$.

Proof: Let $\mathcal{F} = V$ and $\mathcal{S} = E$, with $R(v_1, e) = A$ and $R(v_2, e) = B$ for $e = (v_1, v_2) \in E$.
QED

Thus general SNP assemblies imply no additional constraints on their fragment conflict graphs.

The MAX Induced Bipartite Subgraph problem is: for a given graph $G = (V, E)$ find the largest subset of nodes, $\overline{V} \in V$, such that $G|_{\overline{V}}$ is bipartite. If we let $G = G_{\mathcal{F}}$ and identify $\overline{V}$ with $\overline{\mathcal{F}}$, the nodes retained in MFR, the equivalence is clear.

The reduction to MSR from MAXCUT requires the observation that the edge-induced subgraph of a cut is a bipartite subgraph. Following the construction of the generality lemma, the SNP assembly $\mathcal{A}$ constructed such that $G_{\mathcal{F}} = G$ has the property that the removal of a SNP in $\mathcal{A}$ removes an edge in G. Thus, one can produce a maximal cut by taking the cut $\mathcal{S} - \overline{\mathcal{S}}$. The reduction to MEC requires the observation that for $\mathcal{A}$ constructed according to the lemma, for a graph G, the removal of an edge in G corresponds to changing one A to B or vice versa.

Some of the optimisation problems simplify with additional assumptions:

Theorem 3 *When the SNP matrix has the consecutive 1s property, then the MFR, MSR and MISR problems are polynomial.*

Sketch
We assume that there are no fragments that are contained in other fragments, ie $\forall f_1, f_2 \in \mathcal{F}, \exists s \in \mathcal{S}$: $(R(f_2, s) \neq O) \wedge (R(f_1, s) \neq R(f_2, s))$, and that $\mathcal{S}$ has been given a consecutive 1s ordering. Let $F(f) \in \mathcal{S}$ be the first SNP for which $R(f, B(f)) \neq O$ and let $L(f) \in \mathcal{S}$ be the last SNP for which $R(f, B(f)) \neq O$.

We construct a directed graph on the fragments $\mathcal{D} = (\mathcal{F}, E)$ with edges $E = \{(f_1, f_2) | F(f_1) \leqslant F(f_2) \wedge (R(f_1, s) = R(f_2, s), \forall F(f_2) \leqslant s \leqslant L(f_1))\}$, ie an arc from f_1 to f_2 exists if f_2 starts no earlier than f_1 and has no mismatches with f_1 over their non-O range.

It is clear that a solution to MFR, $\overline{\mathcal{F}} = H_1 \cup H_2$, is a pair of node-disjoint

paths H_1, H_2 such that $|H_1| + |H_2|$ is maximised. In order to find such paths we turn $\mathscr{D}$ into an instance of a maximum cost flow problem, where the cost of the flow is equal to the number of nodes visited. This can be solved in polynomial time. MISR can also be solved as a maximum cost flow problem on $\mathscr{D}$ if the edges are given weights corresponding to the number of SNPs they add to the path.

For MSR, one can first prove theorem 1. A solution of MSR is a maximum independent set of $G_{\mathscr{F}}$. For perfect graphs, a maximum independent set can be found in polynomial time.
QED.

On real data we often can obtain optimal solutions with efficient search methods

In practice and in simulation, our data are not generally C1P, so we have used 'branch-and-bound' algorithms and more sophisticated 'branch-and-cut' algorithms, obtaining optimal results very often. Our experimental data came from the fragments which were used by Celera in its human assembly,[1] and our simulated data were the result of simulated shotgun fragments (using the program described in Myers[22]), aligned in their correct assembly for some typical public sequences. The data are generally not C1P, and thus one would not expect optimal results from our algorithms. It remains open whether there is a less restrictive condition than C1P that gives polynomial complexity and explains why such algorithms work so well.

Multiple optima

We find that the number of optima can be very large

What is a greater issue than the computational complexity of SNP assembly problems is the myriad optima that can result. It has been our observation that producing an optimum solution to a SNP assembly problem is insufficient to deliver relevant insights into the data, and what seems to deliver good insight are statements relevant to *all* optima.

Consider a feasible SNP assembly $\mathscr{A} = (\mathscr{S}, \mathscr{F}, \mathscr{R})$ where $G_{\mathscr{F}}$ has multiple graph components, $G_{\mathscr{F}} = \bigcup_{i=1}^{K} G_i$ where K is the number of components. Since $G_{\mathscr{F}}$ is bipartite, the components G_i are all bipartite, with shores H_{1i} and H_{2i}. From these, one can construct 2^K haplotypings of $\mathscr{A}$ by $H_1 = \bigcup_{i=1}^{K} H_{p_i} i$ and $H_2 = \bigcup_{i=1}^{K} H_{(3-p_i)} i$ where the *phasing vector* $(p_1, p_2, \ldots, p_K)$ is an arbitrary element of $\{1, 2\}^K$. Since most SNP assembly problems in practice have multicomponent $G_{\mathscr{F}}$ or multicomponent $\overline{G}_{\mathscr{F}}$ arise in the optima for nearby feasible $\bar{\mathscr{A}}$, this uniqueness up to phase is a shortcoming of the data that cannot be removed without additional intervention, without which fragments can only be assigned *local haplotypes* identified with the $H_{\{1,2\}i}$.

Multicomponent-based degeneracies are not the only source of non-uniqueness. There can be an exponential number of nearby feasible $\bar{\mathscr{A}}$ with significantly different characteristics. For an infeasible SNP assembly $\mathscr{A}$, there may be many optimal nearby $\bar{\mathscr{A}}$. Our example instance of MFR from Figure 2 has five nearby optima corresponding to the removal of $\{3, 7\}$, or 3 and one of $\{8, 9\}$, or 7 and one of $\{1, 2\}$, as illustrated in Figure 3. As we see here, the degeneracy is not simply a question of independently breaking odd cycles, but there is a coupling to the removals which confounds a simple representation of the solutions beyond an enumeration of each case.

There are conservative strategies that one may apply to MFR in order to obtain consensus results from multiple optima. One is to determine a conservatively reduced fragment set $\hat{\mathscr{F}} = \bigcap \overline{\mathscr{F}}$, where the intersection is taken over all optimal solutions $\overline{\mathscr{F}}$, effectively removing all optimal removal sets from $\mathscr{F}$. The resulting SNP assembly $\hat{\mathscr{A}}$ would be feasible, but probably not optimal. It is useful for inferring those characteristics common to all $\bar{\mathscr{A}}$. This is illustrated in Figure 3 where these fragments are highlighted. From such a presentation, one is made aware that $\{1, 2, 3, 7, 8, 9\}$ are suspicious and that, in every optimal removal, the fragments $\{4, 6\}$ segregate together and away from fragment 5. In

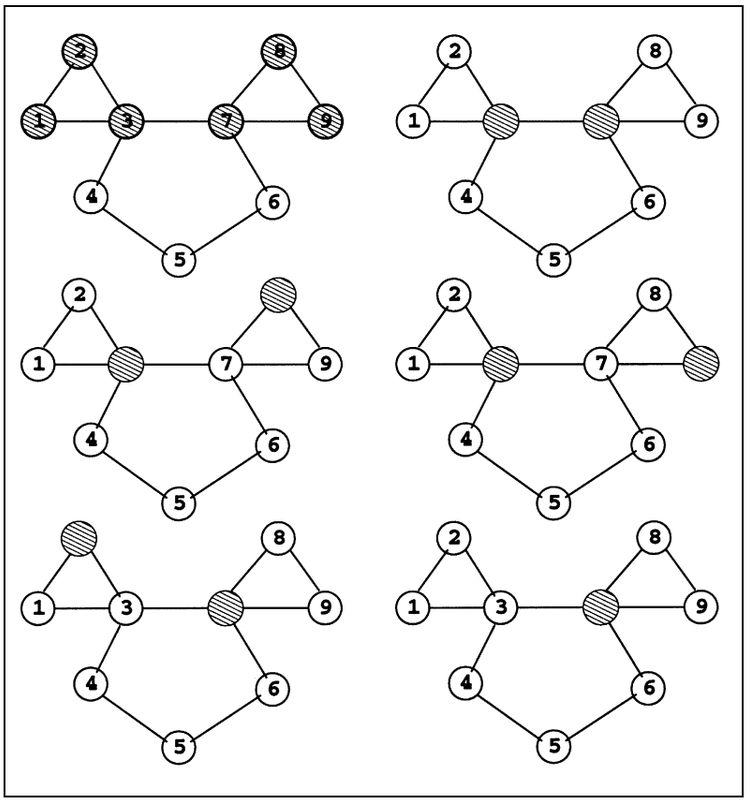

Figure 3: The example SNP matrix, with 'suspicious' fragments highlighted, and an enumeration of the optimal removals

A conservative consensus can be obtained without having to enumerate the optima

the case of MFR, computing the intersection of the optima can be done indirectly, avoiding the total enumeration.

In order to produce inferences based on the SNP data that are insensitive to the arbitrary choice of optima made by an implementation of the optimisation, it is necessary to produce some conservative post-processed result. We consider the correct way to construct such results an open problem.

Branch-and-bound algorithm

We have formulated the above ideas into a branch-and-bound algorithm for finding the intersection of all optimal solutions for MFR. The core of the algorithm is described by the following pseudo-code:

```
graph G
node_set D ← ∅
score s ← optima(∅, ∞)
for each node v in G
   if (optima({v}, ∞) = s)
      D ← D + v
   return G − D
where,
   score s = function optima (node_set R,
   score b)
      if |R| ⩾ b
         return ∞
      else if G − R has an odd length
                                  cycle C
         n = b
         for each node v in C
            n ← min{n, optima(R + v,b)}
         return n
      else
         return |R|
```

The central function, optima, finds the smallest number of nodes one can remove from G to make it bipartite. It finds this number by searching the tree of possible solutions, aborting any branch of the tree that is too deep to lead to a better optimum than the best it had observed up to that point. It then locates any node v removed on any optimal solution by testing whether initially removing v from G changes the optimal solution. If it does

not, then there must have been an optimal solution in which v was removed. Although this procedure requires exponential time in the number of fragments in the worst case, we have found it in practice to be quite efficient on real and realistic simulated data.

To understand how the algorithm works, consider the example problem illustrated in Figures 2 and 3. When optima is run on all of G, it might first quickly locate a solution of size 3, for example the removal of nodes 1, 3 and 7. At that point, it would look only for size 2 solutions, eventually locating, for example, the solution removing nodes 3 and 7. It would then look only for size 1 solutions, quickly determining that no one node's removal would make the graph bipartite. It would thus give the final answer of 2. The algorithm would then examine the graph with node 1 initially removed. It would determine that it can still find a size 2 solution by removing node 7, showing that node 1 must lie on an optimal solution. Proceeding in this manner for nodes 2 to 9, it would determine that initially removing any of nodes 1, 2, 3, 7, 8 or 9 still results in a size 2 solution and that therefore those nodes each lie in some optimal solution. That would leave us with a core graph of nodes 4, 5 and 6 from which to infer haplotypes.

Several techniques have been explored for accelerating the basic algorithm. An initial upper bound is established by using a greedy graph-colouring heuristic, allowing us to avoid an early search of high-cost solutions. We also experimented with reformulating the graph problem as an integer program (a computationally intractable class of optimisation problem) which can be relaxed to a computationally tractable linear program. The solution to the linear program, run within each call to the optima function, establishes a lower bound on the solution cost which can be used to prune away sub-trees that cannot lead to optimal solutions. While this and similar more sophisticated methods would be expected to be more efficient on difficult problem instances, the simpler methods proved to be considerably faster on real data sets. In fact, an exhaustive enumeration of all possible optimal solutions – achieved by searching the branch-and-bound tree for solutions at least as good as the best seen rather than strictly better than the best seen – tended to be even faster. That simpler methods tended to work better in practice than more sophisticated ones reflects the fact that the real data sets we examined were generally very close to bipartite.

Further work is required to understand the boundary between the tractable and intractable instances of SNP assembly problems

DISCUSSION

A common theme of the theoretical results is that these problems have easy special cases but harder general cases. Even the easiest variants can correspond to real-world problem variants; for example, SNP matrices generated from EST sequences will be C1P and therefore polynomially solvable for the MFR, MSR and MISR problems. Other important problem variants do not, however, fall into the known polynomial subsets of the problem space, and an important avenue of future work will therefore be filling in the gaps between the variants we have already characterised to try to locate further tractable but realistic subsets of the problem space. In practice, we have found heuristic methods for generalised fragment removal to work efficiently in almost all cases when run on real data, suggesting that appropriately formulated models may show the problem to be manageable for all reasonable data. It remains to be seen if a model can be constructed that captures the complications of real data that led to our general formulations but are sufficiently specialised so as to yield theoretically tractable problems.

As the difficulties with 'optimal' solutions reveal, there is room for better formulations of the problem. Our objective functions do not yet fully capture what we consider an ideal solution to the problem, suggesting a need for better definitions. Furthermore, there

are many problem details that could be incorporated into models as we get a better grasp on the simpler variants. In particular, some accounting could be considered of the relative confidences we may have in particular fragments or particular DNA bases of fragments. Furthermore, it may be that we will continue to need multiple variants depending on the source of experimental data (eg shotgun fragments versus EST sequences) and on the goal of the haplotyping project.

Even if we continue to work with hard general cases, there is room for improvement in heuristic methods. To date, we have developed only simple variants of such methods and only for MFR. More sophisticated techniques for the different problem variations may yield solutions that work more efficiently for our specific problems, either in the worst case or in the average case of 'biologically realistic' data, a concept that remains to be rigorously formalised.

A final concern is the need for data. While the methods we describe are intended to work with data fortuitously available as the output of automated sequencers, haplotyping may become important enough to merit data generation methods more specifically tuned to the nature of haplotyping. To do so would require understanding how the nature of fragment data might be changed to improve its suitability for haplotyping and how experimental methods might generate such data.

Acknowledgments

We thank Jinghui Zhang and William Rowe for introducing us to the computational aspects of haplotyping. We also thank Vineet Bafna for his contributions to understanding the complexity of some formalisations. We are grateful to Andy Clark for helpful discussions. Finally, we thank our anonymous reviewers for their suggestions and criticisms.

References

1. Venter, J. C., Adams, M. D., Myers, E. W. *et al.* (2001), 'The sequence of the human genome', *Science*, Vol. 291, pp. 1145–1434.
2. International Human Genome Sequencing Consortium (2001), 'Initial sequencing and analysis of the human genome', *Nature*, Vol. 304, pp. 412–417.
3. Altshuler, D., Pollara, V. J., Cowles, C. R. *et al.* (2000), 'An SNP map of the human genome generated by reduced representation shotgun sequencing', *Nature*, Vol. 407, pp. 513–519.
4. Mullikin, J. C., Hunt, S. E., Cole, C. G. *et al.* (2000), 'An SNP map of human chromosome 22', *Nature*, Vol. 407, pp. 516–520.
5. Sachidanandam, R., Weissman, D., Schmidt, S. C. *et al.* (2001), 'A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms', *Nature*, Vol. 409, pp. 928–933.
6. Stephens, J. C., Schneider, J. A., Tanguay, D. A. *et al.* (2001), 'Haplotype variation and linkage disequilibrium in 313 human genes', *Science*, Vol. 293, pp. 489–493.
7. Orkin, S. H. and Kazazian, H. H. (1984), 'The mutation and polymorphism of the human beta-globin gene and its surrounding DNA', *Annu. Rev. Genet.*, Vol. 18, pp. 131–171.
8. Woo, S. L. C. (1988), 'Collation of RFLP haplotypes at the human phenylalanine hydroxylase (PAH) locus', *Amer. J. Hum. Genet.*, Vol. 43, pp. 781–783.
9. Kreitman, M. (1983), 'Nucleotide polymorphism at the alcohol dehydrogenase locus of *Drosophila melanogaster*', *Nature*, Vol. 304, pp. 412–417.
10. Maeda, N., Bliska, J. B. and Smithies, O. (1983), 'Recombination and balanced chromosome polymorphism suggested by DNA sequences 5′ to the human delta-globin gene', *Proc. Natl Acad. Sci. USA*, Vol. 80, pp. 5012–5016.
11. Ruano, G., Kidd, K. K. and Stephens, J. C. (1990), 'Haplotype of multiple polymorphisms resolved by enzymatic amplification of single DNA molecules', *Proc. Natl Acad. Sci. USA*, Vol. 87, pp. 6296–6300.
12. Clark, A. G. (1990), 'Inference of haplotypes from PCR-amplified samples of diploid populations', *Mol. Biol. Evol.*, Vol. 7(2), pp. 111–122.
13. Gusfield, D. (2000), A practical algorithm for optimal inference of haplotypes from diploid populations, in 'Proceedings of the Eight International Conference on Intelligent Systems for Molecular Biology, La Jolla, CA', AAAI Press, Menlo Park, CA.
14. Excoffier, L. and Slatkin, M. (1995), 'Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population', *Mol. Biol. Evol.*, Vol. 12, pp. 921–927.
15. Fallin, D. and Schork, N. J. (2000), 'Accuracy of haplotype frequency estimation for biallelic

loci, via the expectation-maximization algorithm for unphased diploid genotype data', *Amer. J. Hum. Genet.*, Vol. 67, pp. 947–959.

16. Long, J. C., Williams, R. C. and Urbanek, M. (1995), 'An E-M algorithm and testing strategy for multiple-locus haplotypes', *Amer. J. Hum. Genet.*, Vol. 56, pp. 799–810.
17. Stephens, M., Smith, N. J. and Donnelly, P. (2001), 'A new statistical method for haplotype reconstruction from population data', *Amer. J. Hum. Genet.*, Vol. 68, pp. 978–989.
18. Tishkoff, S. A., Pakstis, A. J., Ruano, G. and Kidd, K. K. (2000), 'The accuracy of statistical methods for estimation of haplotype frequencies: an example from the CD4 locus', *Amer. J. Hum. Genet.*, Vol. 67, pp. 518–522.
19. Sanger, F., Coulson, A. R., Hong, G. F. *et al.* (1982), 'Nucleotide sequence of bacteriophage λ DNA', *J. Mol Biol*, Vol. 162(4), pp. 729–773.
20. Fleischmann, R. D., Adams, M. D., White, O. *et al.* (1995), 'Whole-genome random sequencing and assembly of *Haemophilus influenzae Rd*', *Science*, Vol. 269, pp. 496–512.
21. Lancia, G., Bafna, V., Istrail, S. *et al.* (2001), SNPs problems, complexity, and algorithms, in 'Lecture Notes in Computer Science 2161', European Symposium on Algorithms, Aarhus, Denmark, Springer, Berlin, pp. 182–193.
22. Myers, E. W. (1999), A dataset generator for whole genome shotgun sequencing, in 'Conference on Intelligent Systems for Molecular Biology (ISMB 99)', AAAI Press, Menlo Park, CA, pp. 202–210.